



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

PENGEMBANGAN KOMPONEN UPLOADING MEDIA UNTUK Mendukung *ELEARNING* PADA JARINGAN DINAMIS

RAHMAT DANIYANTO
5111100149

Dosen Pembimbing I
WASKITHO WIBISONO, S.Kom., M.Eng., Ph.D.

ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - KI141502

PENGEMBANGAN MEDIA UPLOADING UNTUK MENDUKUNG E-LEARNING PADA KONDISI JARINGAN DINAMIS

RAHMAT DANIYANTO
5111100149

Dosen Pembimbing I
WASKITHO WIBISONO, S.Kom., M.Eng., Ph.D.
ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya, 2017

(halaman ini sengaja dikosongkan)

FINAL PROJECT - KI141502

**DEVELOPMENT UPLOADING MEDIA
COMPONENT ON E-LEARNING FOR DYNAMIC
NETWORK CONDITIONS**

RAHMAT DANIYANTO
5111100149

Supervisor
WASKITHO WIBISONO, S.Kom., M.Eng., Ph.D.
ROYYANA MUSLIM IJTIHADIE, S.Kom., M.Kom

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
Sepuluh Nopember Institute of Technology
Surabaya, 2017

(halaman ini sengaja dikosongkan)

LEMBAR PENGESAHAN
PENGEMBANGAN KOMPONEN MEDIA UPLOADING
UNTUK Mendukung E-ELEARNING PADA KONDISI
JARINGAN DINAMIS

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
Pada
Bidang Studi Komputasi Berbasis Jaringan
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
Rahmat Daniyanto
NRP: 5111 100 149

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Waskito Wibisono, S.Kom., M.Ts., Ph.D.
NIP: 19741022 200003 1 001 (Pembimbing 1)
2. Royyana Muslim Ijtihadie, S.Kom., M.Kom.
NIP: 19770824 200604 1 001 (Pembimbing 2)

SURABAYA

JUNI 2017

(halaman ini sengaja dikosongkan)

PENGEMBANGAN KOMPONEN MEDIA UPLOADING UNTUK Mendukung E-ELEARNING pada kondisi JARINGAN DINAMIS

Nama Mahasiswa : Rahmat Daniyanto
NRP : 5111 100 149
Jurusan : Teknik Informatika FTIF ITS
**Dosen Pembimbing 1 : Waskito Wibisono, S.Kom., M.Eng.,
Ph.D.**
**Dosen Pembimbing 2 : Royyana Muslim Ijtihadie,
S.Kom.,M.Kom**

Abstrak

Kemajuan teknologi informasi saat ini memudahkan setiap orang dalam kegiatannya masing-masing, Saat ini kegiatan belajar mengajar tidak harus dilakukan dengan cara tatap muka langsung tetapi dapat dilakukan dengan melalui secara *online* melalui internet. Pengajar dan murid dapat berkomunikasi melalui suatu web yang didesain untuk memudahkan kegiatan belajar mengajar. Saat ini masyarakat dapat mengakses internet melalui media seperti komputer , laptop serta *smartphone* yang sudah banyak digunakan.

Pada web terdapat banyak fungsi yang dapat digunakan oleh pengguna untuk berdiskusi antar pengajar dan murid . Salah satunya pengajar dapat memberikan tugas pada murid. Pengajar dapat membuat topik bahasan tertentu lalu murid mengumpulkan tugas yang dikerjakan dengan cara mengunggahnya. Namun kualitas koneksi internet tidak

selalu baik, ada kalanya koneksi internet yang buruk mengakibatkan proses mengunggah tugas menjadi gagal.

Dalam tugas akhir ini , *elearning* akan digunakan sebagai sarana pengumpulan tugas.. Dalam proses mengunggah , web akan menggunakan *Chunked Upload Progress* yang memungkinkan pengguna untuk mengatasi masalah koneksi yang buruk saat mengupload *Upload Progress* akan tersimpan saat proses mengunggah sedang berlangsung pada kondisi koneksi jaringan yang dinamis

Kata kunci: Adaptif, jaringan dinamis, e-learning , Chunked Upload Progress

DEVELOPMENT UPLOADING MEDIA COMPONENT ON E-LEARNING FOR DYNAMIC NETWORK CONDITIONS

Student's Name : Rahmat Daniyanto
Student's ID : 5111 100 149
Department : Informatics Engineering, FTIF-ITS
First Advisor : Waskito Wibisono, S.Kom., M.Eng., Ph.D.
Second Advisor : Royyana Muslim Ijtihadie, S.Kom., M.Kom

Abstract

Advances in information technology today facilitate everyone in their respective activities, currently teaching and learning activities should not be done by face to face but can be done through online through the internet. Teachers and students can communicate through a web designed to facilitate teaching and learning activities. Currently people can access the internet through media such as computers, laptops and smartphones that have been widely used

On the web , there are many functions that can be used by users to discuss between teachers and students. One of the teachers can give assignments to the students. Teachers can create specific topics and students collect tasks done by uploading them. But the quality of internet connection is not always good, there are times when a bad internet connection resulted in the process of uploading tasks to fail.

In this final project, elearning will be used as a means of collecting tasks. In the process of uploading, the web will use Chunked Upload Progress which allows users to solve bad

connection problems while uploading.Upload Progress will be saved while uploading is in progress under dynamic network connection conditions

Keywords: Adaptive, dynamic network, e-learning,Chunked upload

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan anugerah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Pengembangan Komponen *Uploading Media* untuk Mendukung *E-learning* Pada Kondisi Jaringan Dinamis”.

Harapan dari penulis semoga apa yang tertulis di dalam buku Tugas Akhir ini dapat bermanfaat bagi pengembangan ilmu pengetahuan saat ini, serta dapat memberikan kontribusi yang nyata bagi kampus Teknik Informatika, ITS, dan bangsa Indonesia.

Dalam pelaksanaan dan pembuatan Tugas Akhir ini tentunya sangat banyak bantuan yang penulis terima dari berbagai pihak, tanpa mengurangi rasa hormat penulis ingin mengucapkan terima kasih sebesar-besarnya kepada:

1. Allah SWT yang telah memberikan kesehatan dan kemampuan kepada penulis untuk menyelesaikan Tugas Akhir ini.
2. Kedua orang tua penulis yaitu Bapak Soedarko dan Ibu Ernita yang tak henti-hentinya memberikan semangat dan kasih sayang serta selalu memberikan doa kepada penulis.
3. Saudara kandung penulis yaitu Rahmat Septian Dendi yang telah memberikan motivasi agar penulis lulus tepat waktu.
4. Bapak Waskito Wibisono, S.Kom., M.Eng., Ph.D. selaku dosen pembimbing penulis yang selalu meluangkan waktu dan memberikan kepercayaan, dukungan, nasihat, serta semangat kepada penulis.
5. Bapak Tohari Ahmad., M.Sc. selaku dosen wali penulis yang telah membimbing penulis selama kuliah di Teknik Informatika ITS.
6. Teman-teman seperjuangan Faris ,Adi dan Irfan yang sedang bersama-sama mengerjakan Tugas Akhir.
7. Juga kepada semua pihak yang belum sempat disebutkan satu per satu yang telah membantu penulis untuk menyelesaikan Tugas Akhir ini.

Kesempurnaan tentu masih jauh tercapai pada Tugas Akhir ini, sehingga penulis mengharapkan saran dan kritik yang membangun dari pembaca untuk perbaikan ke depan. Semoga Tugas Akhir ini dapat bermanfaat bagi perkembangan ilmu pengetahuan dan bagi semua pihak.

Surabaya, Juni 2017

Rahmat Daniyanto

DAFTAR ISI

| | |
|---|-------------------------------------|
| LEMBAR PENGESAHAN..... | Error! Bookmark not defined. |
| Abstrak | ix |
| Abstract | xi |
| KATA PENGANTAR..... | xiii |
| DAFTAR ISI | xv |
| DAFTAR GAMBAR | xvii |
| DAFTAR TABEL | xix |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang..... | 1 |
| 1.2 Rumusan Permasalahan | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan dan Manfaat..... | 3 |
| 1.5 Metodologi | 3 |
| 1.6 Sistematika Penulisan | 5 |
| BAB II TINJAUAN PUSTAKA..... | 7 |
| 2.1 Pembelajaran Kolaboratif | 7 |
| 2.2 SATC..... | 10 |
| 2.3 MySQL | 11 |
| 2.4 PHP..... | 12 |
| 2.5 JavaScript | 12 |
| 2.6 Asynchronous JavaScript and XML (AJAX) | 13 |
| 2.7 Kondisi Jaringan Dinamis | 14 |
| BAB III PERANCANGAN PERANGKAT LUNAK | 17 |
| 3.1 Deskripsi Umum Sistem..... | 17 |
| 3.2 Arsitektur Umum Sistem | 18 |
| 3.3 Perancangan Diagram Alir Data Aplikasi | 19 |
| 3.3.1 Perancangan Diagram Konteks Aplikasi | 19 |
| 3.3.2 Diagram Alir Data Level 0 Aplikasi..... | 20 |
| 3.4 Diagram Alir Aplikasi | 21 |
| 3.4.1 Diagram Alir Proses Inisialisasi Koneksi klien | 21 |
| 3.4.2 Diagram Alir Proses Estimasi Kondisi Jaringan | 22 |
| 3.4.3 Diagram Alir Proses Eksekusi Adaptasi Uploading | 23 |
| 3.5 Perancangan Database | 24 |

| | | |
|--|--|----|
| 3.5.1 | <i>Tabel Kelas_tugas_pengumpulan</i> | 24 |
| 3.5.2 | <i>Tabel kelas_tugas_pengumpulan_chunk</i> | 26 |
| 3.6 | Rancangan Antarmuka Sistem..... | 27 |
| 3.6.1 | <i>Tampilan Tugas(semua /belum selesai/sudah selesai)</i> | 27 |
| 3.6.2 | <i>Tampilan Upload Tugas</i> | 27 |
| 3.6.3 | <i>Tampilan Upload melalui Kamera</i> | 28 |
| 3.6.4 | <i>Tampilan Melanjutkan Upload</i> | 29 |
| 3.6.5 | <i>Tampilan Daftar File</i> | 30 |
| BAB IV IMPLEMENTASI PERANGKAT LUNAK | | 32 |
| 4.1 | Lingkungan Pembangunan Perangkat Lunak | 33 |
| 4.1.1 | <i>Lingkungan Perangkat Lunak</i> | 33 |
| 4.1.2 | <i>Lingkungan Perangkat Keras</i> | 33 |
| 4.2 | Mekanisme Perangkat Lunak | 33 |
| 4.2.1 | <i>Web Server</i> | 34 |
| 4.2.2 | <i>Implementasi Fungsi</i> | 34 |
| 4.3 | Implementasi Antarmuka Perangkat Lunak | 42 |
| 4.3.1 | <i>Tampilan Awal</i> | 42 |
| 4.3.2 | <i>Tampilan tugas guru</i> | 43 |
| 4.3.3 | <i>Tampilan awal tugas murid</i> | 44 |
| 4.3.4 | <i>Tampilan Tambah Tugas</i> | 44 |
| 4.3.5 | <i>Tampilan Upload Tugas</i> | 45 |
| 4.3.6 | <i>Tampilan Daftar File</i> | 46 |
| 4.3.7 | <i>Tampilan Lanjut Upload</i> | 47 |
| BAB V UJI COBA DAN EVALUASI | | 49 |
| 5.1 | Lingkungan Uji Coba | 49 |
| 5.2 | Data Uji Coba | 50 |
| 5.3 | Skenario Uji Coba | 50 |
| 5.3.1 | <i>Uji Coba Fungsionalitas</i> | 51 |
| 5.3.2 | <i>Uji Coba Performa</i> | 66 |
| BAB VI PENUTUP | | 73 |
| LAMPIRAN | | 75 |
| DAFTAR PUSTAKA | | 81 |
| BIODATA PENULIS | | 83 |

DAFTAR GAMBAR

| | |
|--|----|
| GAMBAR 2.1 STRUKTUR SISTEM PEMBELAJARAN KOLABORATIF [4]. | 10 |
| GAMBAR 2.2 PERBANDINGAN ANTARA APLIKASI WEB KLASIK DENGAN APLIKASI WEB BERBASIS AJAX [13] | 14 |
| GAMBAR 3.1 GAMBARAN UMUM ALUR KERJA APLIKASI | 18 |
| GAMBAR 3.2 DIAGRAM KONTEKS APLIKASI | 20 |
| GAMBAR 3.3 DIAGRAM ALIR DATA LEVEL 0 | 21 |
| GAMBAR 3.4 DIAGRAM ALIR KONEKSI KLIEN | 22 |
| GAMBAR 3.5 DIAGRAM ALIR ESTIMASI KONDISI JARINGAN | 23 |
| GAMBAR 3.6 DIAGRAM ALIR EKSEKUSI ADAPTASI UPLOADING | 24 |
| GAMBAR 3.7 TABEL KELAS_TUGAS_PENGUMPULAN | 24 |
| GAMBAR 3.8 TABEL KELAS_TUGAS_PENGUMPULAN_CHUNK | 26 |
| GAMBAR 3.9 DESAIN ANTARMUKA TAMPILAN WEB | 27 |
| GAMBAR 3.10 DESAIN ANTARMUKA TAMPILAN <i>UPLOAD FILE</i> | 28 |
| GAMBAR 3.11 DESAIN ANTARMUKA TAMPILAN <i>UPLOAD MELALUI KAMERA</i> | 29 |
| GAMBAR 3.12 DESAIN ANTARMUKA TAMPILAN MELANJUTKAN <i>UPLOAD</i> | 30 |
| GAMBAR 3.13 DESAIN ANTARMUKA TAMPILAN DAFTAR <i>FILE</i> | 31 |
| GAMBAR 4.1 HALAMAN AWAL <i>ELEARNING</i> | 34 |
| GAMBAR 4.2 IMPLEMENTASI MENGECEK KECEPATAN | 35 |
| GAMBAR 4.3 IMPLEMENTASI FUNGSI ADAPTIF PEMOTONGAN <i>FILE</i> | 36 |
| GAMBAR 4.4 IMPLEMENTASI FUNGSI PENGIRIMAN <i>FILE</i> (1) | 37 |
| GAMBAR 4.5 IMPLEMENTASI FUNGSI PENGIRIMAN <i>FILE</i> (2) | 38 |
| GAMBAR 4.6 IMPLEMENTASI FUNGSI MELANJUTKAN PENGIRIMAN <i>FILE</i> | 39 |
| GAMBAR 4.7 IMPLEMENTASI FUNGSI MENGAMBIL GAMBAR DARI KAMERA | 40 |
| GAMBAR 4.8 IMPLEMENTASI FUNGSI MENGIRIM <i>FILE</i> DARI KAMERA | 41 |
| GAMBAR 4.9 TAMPILAN AWAL MURID | 42 |

| | |
|--|----|
| GAMBAR 4.10 TAMPILAN AWAL GURU | 43 |
| GAMBAR 4.11 TAMPILAN AWAL TUGAS GURU..... | 43 |
| GAMBAR 4.12 TAMPILAN AWAL TUGAS MURID | 44 |
| GAMBAR 4.13 TAMPILAN TAMBAH TUGAS | 45 |
| GAMBAR 4.14 TAMPILAN UPLOAD TUGAS MELALUI <i>FILE</i> | 46 |
| GAMBAR 4.15 TAMPILAN UPLOAD TUGAS MELALUI KAMERA .. | 46 |
| GAMBAR 4.16 TAMPILAN DAFTAR <i>FILE</i> | 47 |
| GAMBAR 4.17 TAMPILAN LANJUT UPLOAD | 47 |
| GAMBAR 5.1 HASIL UJI COBA MENGECEK KECEPATAN..... | 52 |
| GAMBAR 5.2 HASIL UJI COBA MENGECEK KECEPATAN MELALUI <i>CONSOLE</i> | 53 |
| GAMBAR 5.3 INFORMASI <i>FILE</i> BERTIPE RAR | 54 |
| GAMBAR 5.4 PROSES PENGIRIMAN <i>FILE</i> PADA UJICOB PENGIRIMAN <i>FILE</i> 1 | 55 |
| GAMBAR 5.5 PROSES PENGIRIMAN <i>FILE</i> SELESAI PADA UJICOB PENGIRIMAN <i>FILE</i> 1 | 56 |
| GAMBAR 5.6 HASIL UJI COBA MENGIRIM <i>FILE</i> 1 MELALUI <i>CONSOLE</i> | 57 |
| GAMBAR 5.7 INFORMASI <i>FILE</i> BERTIPE PNG..... | 58 |
| GAMBAR 5.8 PROSES PENGIRIMAN TUGAS MELALUI <i>FILE</i> PADA UJICOB PENGIRIMAN <i>FILE</i> 2 | 60 |
| GAMBAR 5.9 PROSES PENGIRIMAN TUGAS MELALUI <i>FILE</i> SELESAI PADA UJICOB PENGIRIMAN <i>FILE</i> 2 | 61 |
| GAMBAR 5.10 HASIL UJI COBA MENGIRIM TUGAS MELALUI <i>FILE</i> 2 MELALUI <i>CONSOLE</i> | 61 |
| GAMBAR 5.11 HASIL UJI COBA PROSES MENGIRIM GAMBAR MELALUI KAMERA PERANGKAT | 63 |
| GAMBAR 5.12 HASIL UJI COBA PROSES SELESAI MENGIRIM GAMBAR MELALUI KAMERA PERANGKAT | 63 |
| GAMBAR 5.13 <i>FILE</i> YANG BELUM SELESAI TERKIRM | 65 |
| GAMBAR 5.14 <i>FILE</i> YANG DIPILIH TIDAK SAMA DENGAN <i>FILE</i> YANG DIMINTA | 65 |

DAFTAR TABEL

| | |
|--|----|
| TABEL 2.1 MODEL PEMBELAJARAN KOLABORATIF [4] | 7 |
| TABEL 5.1 SKENARIO KONDISI JARINGAN | 50 |
| TABEL 5.2 PROSEDUR UJI COBA MENGECEK KECEPATAN | 51 |
| TABEL 5.3 PROSEDUR UJI COBA PENGIRIMAN <i>FILE</i> 1 | 54 |
| TABEL 5.4 PROSEDUR UJI COBA PENGIRIMAN TUGAS MELALUI <i>FILE</i> 2 | 58 |
| TABEL 5.5 PROSEDUR UJI COBA PENGIRIMAN TUGAS MELALUI KAMERA | 62 |
| TABEL 5.6 PROSEDUR UJI COBA MELANJUTKAN PENGIRIMAN TUGAS | 64 |
| TABEL 5.7 HASIL UJI COBA PERFORMA DALAM KEADAAN NORMAL | 67 |
| TABEL 5.8 HASIL UJI COBA PERFORMA DALAM KEADAAN REGULAR 3G | 68 |
| TABEL 5.9 HASIL UJI COBA PERFORMA DALAM KEADAAN GPRS | 69 |
| TABEL 5.10 HASIL UJI COBA UPLOAD MENGGUNAKAN METODE CHUNK <i>FILE</i> | 70 |
| TABEL 5.11 HASIL UJI COBA UPLOAD MENGGUNAKAN METODE UPLOAD PHP | 71 |

(halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

Pada bab ini akan dijelaskan mengenai beberapa hal dasar dalam Tugas Akhir ini yang meliputi latar belakang, rumusan masalah, tujuan dan manfaat pembuatan Tugas Akhir, serta metodologi dan sistematika penulisan buku Tugas Akhir ini.

1.1 Latar Belakang

Kegiatan belajar mengajar secara *online* yaitu *e-learning* telah dilakukan beberapa perguruan tinggi di Indonesia. Dengan *e-learning* tersebut pengajar dan murid dapat berinteraksi secara langsung. *E-learning* memudahkan pengajar dan murid untuk bertatap muka dengan tidak perlu bertemu untuk kegiatan belajar mengajar, tetapi hanya cukup menggunakan media seperti *smartphone* atau komputer.

Tantangan yang ada sekarang ini yaitu bagaimana mengatur kualitas pengiriman multimedia tersebut sesuai dengan kondisi *traffic* jaringan internet. Seperti yang diketahui, kondisi jaringan internet tidak selalu lancar. Banyak sekali hambatan yang menyebabkan kondisi jaringan tidak stabil. Untuk komunikasi *real time* membutuhkan pengaturan terhadap multimedia sesuai dengan kondisi jaringan agar mengantisipasi adanya delay yang terlalu lama sehingga mengurangi kualitas dari layanan komunikasi tersebut.

Kondisi jaringan dinamis adalah keadaan suatu koneksi jaringan yang berubah-ubah sesuai dengan kondisi *traffic* jaringan. Keadaan yang berubah-ubah tersebut yang menyebabkan pengiriman paket pada jaringan memerlukan waktu yang berbeda-beda. Semakin baik kondisi *traffic* jaringan, maka semakin baik pula pengiriman yang dilakukan. Sebaliknya jika kondisi *traffic* jaringan jelek, maka semakin lambat pula pengiriman yang dilakukan.

Dalam tugas akhir ini , *plugin* pada Content Management System (CMS)[1] *elearning* akan digunakan sebagai sarana pengumpulan tugas.. Dalam proses mengunggah , web akan menggunakan *Chunked Upload Progress*[2] yang memungkinkan pengguna untuk mengatasi masalah koneksi yang buruk saat mengupload *Upload Progress* akan tersimpan saat proses mengunggah sedang berlangsung pada kondisi koneksi jaringan yang dinamis

1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat aplikasi belajar mengajar yang kolaboratif ?
2. Bagaimana cara mengatasi koneksi internet yang lambat saat murid akan melakukan pengumpulan tugas?
3. .Bagaimana cara mengatasi koneksi internet yang terputus saat murid akan melakukan pengumpulan tugas?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Aplikasi ini berbasis web dengan bahasa pemrograman PHP.
2. Uji coba dilakukan pada web browser yang mendukung sistem layanan multimedia yaitu Google Chrome 23 (atau versi yang lebih baru).
3. Uji coba dilakukan pada jaringan lokal *Ethernet* dan Wi-Fi.

1.4 Tujuan dan Manfaat

Tujuan dari pembuatan tugas akhir ini yaitu membuat aplikasi yang dapat membantu pengajar dan murid dalam melakukan komunikasi dalam pengerjaan suatu tugas pada kondisi jaringan dinamis.

Manfaat dari hasil pembuatan tugas akhir ini yaitu Mempermudah pengajar dan murid dalam proses belajar mengajar. Serta Membantu murid dalam proses pengumpulan tugas pada saat kondisi jaringan dinamis.

1.5 Metodologi

1. Penyusunan Proposal Tugas Akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan untuk pembuatan tugas akhir. Pendahuluan ini terdiri atas latar belakang diajukannya tugas akhir, permasalahan yang diangkat, batasan masalah, tujuan dan manfaat dibuatnya tugas akhir ini. Selain itu juga dijelaskan tentang tinjauan pustaka sebagai referensi pendukung untuk mengerjakan tugas akhir tersebut yaitu *Elearning* dan *Chunked Media Upload*. Pada Sub Bab Metodologi menjelaskan tentang mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Jadwal kegiatan yang dilakukan juga dilampirkan dalam proposal tugas akhir ini agar tepat waktu.

2. Studi Literatur

Pada studi literature ini, yang akan dipelajari untuk pembuatan tugas akhir ini E-learning adalah suatu sistem atau konsep pendidikan yang memanfaatkan teknologi informasi dalam proses belajar mengajar. Pengajar dan murid mampu berkomunikasi dengan multimedia secara *real time* antar pengguna

web browser. Dan untuk mengantisipasi adanya koneksi buruk saat proses pengumpulan tugas, maka diperlukan pengukuran kinerja dari jaringan untuk proses pengunggahan *file*.

3. Analisis dan Desain Perangkat Lunak

Fitur dari aplikasi ini yaitu:

1. Terdapat 2 user yaitu pengajar dan murid.
2. Media pengunggahan *file* yang digunakan untuk pengumpulan tugas.
3. Pengaturan proses pengunggahan *file* berdasarkan kondisi jaringan.

4. Implementasi Perangkat Lunak

Aplikasi ini akan dibangun dengan bahasa pemrograman PHP. Aplikasi ini akan dibangun dengan menggunakan *Integrated Development Environment* (IDE) Sublime Text 2.0.2.

5. Pengujian dan Evaluasi

Pengujian dari aplikasi ini akan dilakukan dengan menggunakan 1 *server* dan 2 user dari 1 komputer. User yang dijadikan pengujian diberikan kondisi jaringan yang dinamis, misal dengan menambahkan packet loss dan bandwidth limit.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
 - a. Latar Belakang
 - b. Rumusan Masalah
 - c. Batasan Tugas Akhir
 - d. Tujuan
 - e. Metodologi
 - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

1.6 Sistematika Penulisan

Buku tugas akhir ini disusun dengan sistematika penulisan sebagai berikut:

1. Bab I. Pendahuluan

Bab ini berisi latar belakang, permasalahan, tujuan, batasan permasalahan, metodologi, dan sistematika penulisan.

2. Bab II. Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar teori penunjang yang digunakan untuk mendukung penyelesaian Tugas Akhir.

3. Bab III. Perancangan Perangkat Lunak

Bab ini berisi tentang perancangan sistem, diagram alir, dan perancangan antarmuka yang akan dibuat.

4. Bab IV. Implementasi Perangkat Lunak

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa *pseudocode* dan *screenshot* aplikasi.

5. Bab V. Evaluasi dan Uji Coba

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian fungsionalitas dan pengujian performa dalam beberapa skenario.

6. Bab VI. Penutup

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Pada bab ini menjelaskan mengenai teori-teori yang berkaitan dengan pengimplementasian perangkat lunak. Bab ini bertujuan untuk memberikan gambaran secara umum mengenai teori serta alat bantu yang digunakan pada implementasi perangkat lunak pada tugas akhir ini.

2.1 Pembelajaran Kolaboratif

Pembelajaran kolaboratif[3] merupakan sebuah metode pembelajaran dimana para murid yang memiliki kemampuan yang berbeda-beda satu sama lain dikumpulkan dalam kelompok kecil untuk menyelesaikan satu permasalahan yang sama.

Dalam pembelajaran kolaboratif, setiap murid akan menjadi pusat baik dalam aktivitas mengajar maupun belajar, dan proses belajar merupakan sebuah proses dimana mereka memahami tentang konsep dan fakta kemampuan dari mereka sendiri [3]. Di dalam pembelajaran ini, mereka akan menjadi pelajar yang aktif karena mereka membentuk pengetahuan dengan cara melakukan interaksi sosial dengan murid lain yang memiliki pengetahuan lebih.

Tabel 2.1 Model Pembelajaran Kolaboratif [4]

| | Waktu sama | Waktu berbeda |
|-------------|---|--|
| Tempat Sama | (a) Pembelajaran dengan bertemu langsung dalam satu ruangan kelas | (b) Pembelajaran dengan berdiskusi di dalam satu |

| | | |
|-------------------|--|--|
| | | ruangan kelas namun dalam waktu yang berbeda |
| Tempat berbeda | © Pembelajaran dalam waktu yang bersamaan namun berada pada jarak jauh menggunakan internet atau teknologi informasi | (d) Pembelajaran menggunakan internet namun tidak dalam waktu yang sama, misalkan menggunakan email, video, atau buletin |

Melihat pada Tabel 2.1, ada empat model pembelajaran kolaboratif yang umumnya ada berdasarkan cara komunikasi antara para pelajar yang dibagi berdasarkan variabel waktu dan tempat. Empat model pembelajaran kolaboratif tersebut dapat dijelaskan sebagai berikut [4].

(a) Waktu sama – Tempat sama

Model ini merupakan model pembelajaran kolaboratif yang paling umum dilakukan, dimana setiap anggota dalam suatu kelompok saling berinteraksi dan berdiskusi satu sama lain dalam waktu dan tempat yang sama, biasanya dalam bentuk pertemuan untuk berdiskusi masalah tertentu.

(b) Waktu berbeda – Tempat sama

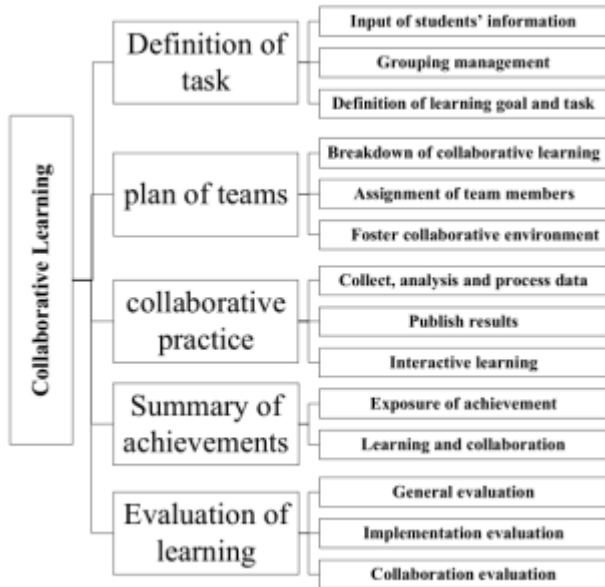
Model kolaborasi ini biasanya dilakukan dengan memberikan informasi tentang pertanyaan atau informasi lainnya ke kelompok dengan menggunakan buletin atau papan tulis.

(c) Waktu sama – Tempat berbeda

Model ini biasanya membutuhkan adanya komunikasi dalam waktu yang sama (Real Time) dalam lingkungan yang berbeda. Alat-alat yang digunakan biasanya meliputi fitur panggilan video, audio, dan media lainnya yang mendukung kemampuan berbagi informasi dan berdiskusi dalam kelompok secara real time.

(d) Waktu berbeda – Tempat berbeda

Model ini pada saat ini adalah yang merupakan paling banyak digunakan karena membutuhkan terjadinya komunikasi dalam waktu yang berbeda dan tempat yang berbeda pula. Alat-alat yang digunakan biasanya berupa surat elektronik, forum, blog, dan lainnya untuk berbagi informasi dan berdiskusi dalam kelompok pelajar.



Gambar 2.1 Struktur Sistem Pembelajaran Kolaboratif [4].

2.2 SATC

SATC[9] - Simple App Task Collecor adalah rintisan aplikasi berbasis web untuk memudahkan pengajar dan murid (ruang lingkup luas, tidak sebatas pembelajaran di dalam kelas) agar lebih mudah dalam kegiatan pemberian dan pengumpulan tugas.

Yang nantinya menjadi target pengguna aplikasi ini adalah:

- Guru (baik guru kelas, tentor, pengajar lembaga terbuka ataupun trainer kegiatan tertentu)
- Peserta didik (baik siswa ataupun siswi pendidikan formal, non-formal, atau hanya event khusus dari organisasi tertentu)
- SATC baik digunakan (optimal) untuk pembelajaran dengan ruang lingkup kecil-sedang/ tidak terlalu banyak peserta didik

SATC berfokus pada manajemen data yang simple dan mudah digunakan untuk berkelanjutan.

- Data yang kami maksud hanyalah :
- Data pengajar
- Data peserta didik
- Data *file* tugas
- Data singkat Subjek kelas dan penugasan

Berikut role untuk admin:

- Menambah pengajar
- Mengedit pengajar
- Menghapus pengajar

Berikut role untuk guru:

- Menambah kelas
- Mengedit kelas
- Menghapus kelas
- Menambahkan siswa ke kelas
- Menghaus siswa dari kelas

Role untuk peserta didik/ siswa

- Peserta didik dapat masuk ke suatu kelas setelah mendaftar sebagai akun peserta didik dan dimasukkan ke kelas oleh guru
- Submit tugas (upload *file*)
- Re-Submit tugas (upload *file* baru untuk menggantikan *file* lama)
- Menghapus *file-file* tugas pribadi yang telah di upload

2.3 MySQL

MySQL [5] adalah sistem manajemen *database* SQL yang bersifat Open Source dan paling populer saat ini. Sistem *database* MySQL mendukung beberapa fitur seperti multithreaded, multi-

user, dan SQL *database* manajemen sistem (DBMS). *Database* ini dibuat untuk keperluan sistem *database* yang cepat, handal dan mudah digunakan.

Pada MySQL terdapat *logical model* yang terdiri dari *databases*, *tables*, *views*, *rows*, dan *columns* untuk membantu penggunaan data yang terstruktur di lingkungan programmer. MySQL juga mendukung sistem klien *server*, sehingga data dapat terpusat di *server* dan dapat diakses oleh banyak klien.

Dalam tugas akhir ini MySQL digunakan untuk manajemen data user sebagai pengguna dari aplikasi WebRTC seperti user, room, dll.

2.4 PHP

PHP [6] merupakan bahasa pemrograman *script server-side* yang didesain untuk pengembangan web. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum. PHP disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer *server*. Hal ini berbeda dibandingkan dengan bahasa pemrograman *client-side* seperti JavaScript yang diproses pada web browser (client).

Kode PHP dapat dicampur dengan kode HTML, atau dapat digunakan dalam kombinasi dengan berbagai mesin template dan kerangka web. Kode PHP biasanya diproses oleh interpreter PHP, yang biasanya diimplementasikan sebagai modul asli web *server* atau *Common Gateway Interface (CGI) executable*. Setelah kode PHP ditafsirkan dan dilaksanakan, *server* web mengirimkan output yang dihasilkan ke klien, biasanya dalam bentuk bagian dari halaman web yang dihasilkan; misalnya, kode PHP dapat menghasilkan halaman Web kode HTML, gambar, atau data lainnya. PHP juga telah berkembang untuk menyertakan *Command Line Interface (CLI)* dan dapat digunakan dalam aplikasi grafis.

2.5 JavaScript

JavaScript [7] merupakan bahasa pemrograman yang umum digunakan dalam pemrograman web. Bahasa pemrograman ini

didukung pada berbagai peramban, salah satunya Google Chrome. Karena sudah umum digunakan, banyak peramban modern berbasis *desktop* maupun *mobile* saat ini menanamkan dukungan *JavaScript*.

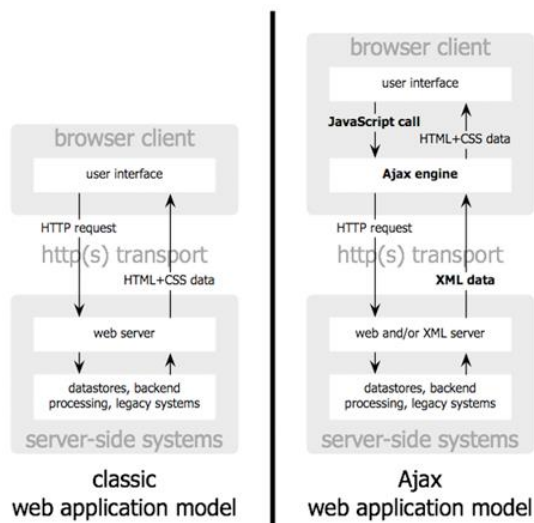
JavaScript merupakan bahasa pemrograman yang berjalan pada sisi klien (*client-side scripting*). Hal ini menyebabkan setiap eksekusi perintah dilakukan oleh peramban dimana pengguna mengakses situs. Penggunaan *JavaScript* sendiri berdampingan dengan HTML dan CSS, dimana *JavaScript* dapat digunakan untuk memanipulasi konten dan desain dari situs.

2.6 Asynchronous JavaScript and XML (AJAX)

AJAX [8] merupakan salah satu teknik pengembangan *website* yang memungkinkan halaman situs dapat menerima konten secara dinamis melalui *server*. Pertukaran data antara pengguna dengan *server* dilakukan secara asinkron, sehingga tidak mengganggu tampilan situs. Implementasi AJAX sendiri memanfaatkan gabungan beberapa komponen, yaitu:

- Situs dengan halaman XHTML dan CSS.
- Tampilan dinamis dan interaktif dengan memanfaatkan *Document Object Model (DOM)*.
- Pertukaran dan manipulasi data menggunakan XML dan XSLT.
- Menerima data secara asinkron menggunakan XMLHttpRequest
- JavaScript untuk menggabungkan seluruh komponen tersebut.

Gambar 2.4 menunjukkan bagaimana perbandingan antara aplikasi *web* klasik dengan aplikasi *web* dengan menggunakan AJAX. Aplikasi *web* tanpa AJAX melakukan permintaan ke *web server* kemudian memberikan respon dalam bentuk halaman HTML, sehingga pengguna akan mengalami perpindahan halaman. Berbeda jika situs mengimplementasi AJAX, permintaan ke *web server* dilakukan melalui perantara JavaScript dengan memanggil objek XMLHttpRequest. *Web server* akan mengirimkan respon berupa data XML yang kemudian diolah untuk ditampilkan ke dalam bentuk HTML dan CSS.



Gambar 2.2 Perbandingan Antara Aplikasi Web Klasik dengan Aplikasi Web Berbasis AJAX [13]

Meskipun objek yang digunakan adalah XMLHttpRequest, namun pada perkembangannya respon yang diolah tidak selalu berupa XML. Saat ini, jenis respon yang populer digunakan adalah dalam format JSON (*JavaScript Object Notation*) yang umumnya digunakan bersamaan dengan jQuery. Selain dalam bentuk teks yang telah terformat, respon juga dapat berupa *plain text* yang dibuat dalam format sesuai keinginan *programmer*.

2.7 Kondisi Jaringan Dinamis

Kondisi jaringan dinamis adalah keadaan suatu koneksi jaringan yang berubah-ubah sesuai dengan kondisi *traffic* jaringan. Keadaan yang berubah-ubah tersebut yang menyebabkan pengiriman paket pada jaringan memerlukan waktu yang berbeda-beda. Kondisi tersebut juga dapat mempengaruhi komunikasi multimedia pada jaringan. Untuk mengantisipasi adanya delay

yang terlalu lama antar paket yang dikirimkan, diperlukan adanya mekanisme untuk mengatur pengiriman paket multimedia. Mekanisme tersebut bisa diterapkan dengan mengukur kinerja jaringan untuk mendapatkan *jitter*, *packet-loss*, dll. Nilai dari pengukuran tersebut kemudian dijadikan acuan untuk mengatur konstrain video dan audio yang akan dikirim.

(Halaman ini sengaja dikosongkan)

BAB III

PERANCANGAN PERANGKAT LUNAK

Pada bab ini akan dijelaskan mengenai dasar perancangan perangkat lunak yang akan dibuat dalam tugas akhir ini. Secara khusus akan dibahas mengenai deskripsi umum aplikasi, perancangan proses, alur, serta gambaran implementasi perangkat lunak.

3.1 Deskripsi Umum Sistem

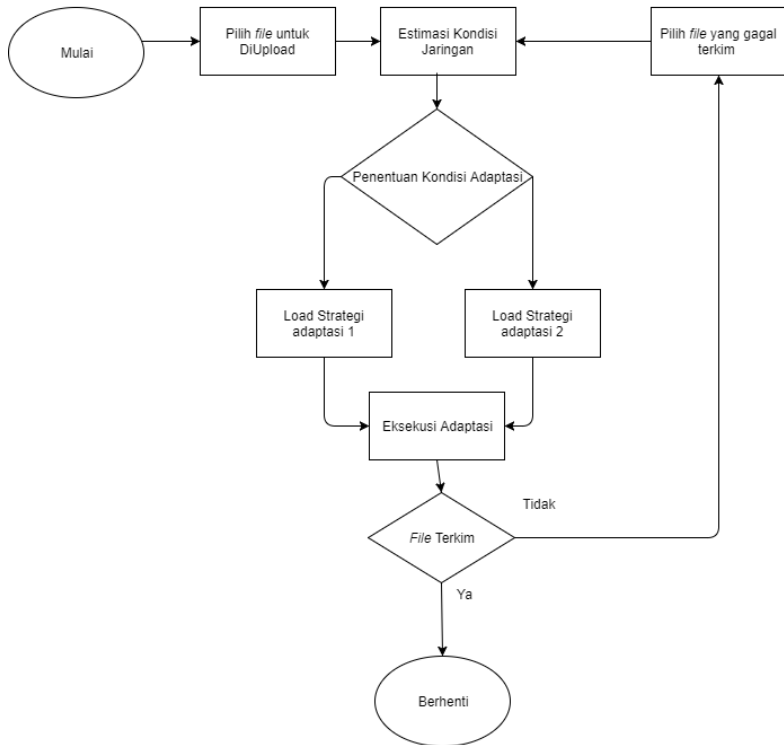
Pada tugas akhir ini dibangun sebuah plugin pada sebuah CMS untuk *elearning* yang memiliki kemampuan untuk mengunggah tugas yang dikerjakan murid yang adaptif terhadap keadaan jaringan saat proses mengunggah dilakukan. Hasil akhir aplikasi web ini dapat beradaptasi dengan keadaan jaringan normal, lambat maupun terputusny koneksi saat murid melakukan proses pengumpulan tugas.

Proses kerja modul ini bermula dengan pengecekan kecepatan upload jaringan pengguna ,yang nantinya akan berpengaruh dalam proses pengiriman *file*. koneksi jaringan terbagi menjadi baik dan buruk. Apabila koneksi baik maka proses pengiriman *file* akan dilakukan dengan keadaan normal sedangkan apabila keadaan jaringan buruk ukuran *chunked file* akan diubah menyesuaikan dengan kecepatan upload pengguna.

Apabila dalam proses pengiriman *file* terjadi gangguan koneksi yang menyebabkan terputusny koneksi dengan *server* ,maka pengguna dapat melanjutkan proses pengiriman *file* tersebut dengan syarat nama *file* dan ukuran *file* yang dikirim sama dengan *file* yang dikirim sebelumnya.

3.2 Arsitektur Umum Sistem

Agar dapat menjalankan fungsinya, maka alur kerja dari kesatuan aplikasi ini dirancang seperti pada Gambar 3.1.



Gambar 3.1 Gambaran Umum Alur Kerja Aplikasi

Berdasarkan Gambar 3.1, alur kerja aplikasi ini dijabarkan sebagai berikut:

1. klien melakukan inisialisasi koneksi peer dengan mengirimkan identitas yang telah didaftarkan

- sebelumnya kepada *server*. *Server* akan mengecek identitas dari klien pada database
2. Ketika klien ingin mengunggah *file* ke aplikasi ,klien terlebih dahulu memilih letak *file* yang akan dikirim.
 3. Lalu aplikasi mengecek kondisi Jaringan , apabila kondisi jaringan sedang buruk maka akan masuk strategi adaptasi 1 sedangkan apabila kondisi jaringan baik akan masuk strategi adaptasi 2.
 4. Strategi adaptasi 1 terjadi saat jaringan buruk. *File* yang akan dikirim akan disimpan sementara pada folder pada device klien . Proses pengunggahan akan diulang saat koneksi jaringan telah baik
 5. Strategi adaptasi 2 terjadi saat kondisi jaringan baik. *File* akan langsung diunggah ke Web
 6. Apabila saat proses jaringan terputus maka klien dapat melanjutkan proses mengirimkan *file* dengan mengirim *file* dengan nama dan ukuran yang sama ,apabila *file* berhasil terkirim maka proses akan berhenti

3.3 Perancangan Diagram Alir Data Aplikasi

Pada bagian ini akan dibahas mengenai gambaran aliran data dan fungsionalitas sistem secara umum. Hal ini direpresentasikan berupa diagram konteks dan diagram alir data level 0

3.3.1 Perancangan Diagram Konteks Aplikasi

Diagram konteks merupakan diagram alir yang menggambarkan sistem secara umum. Semua aktor eksternal serta aliran data masuk dan keluar sistem digambarkan dalam satu diagram, dimana keseluruhan sistem digambarkan dalam satu proses. Konteks diagram aplikasi ini ditunjukkan pada Gambar 3.2.



Gambar 3.2 Diagram Konteks Aplikasi

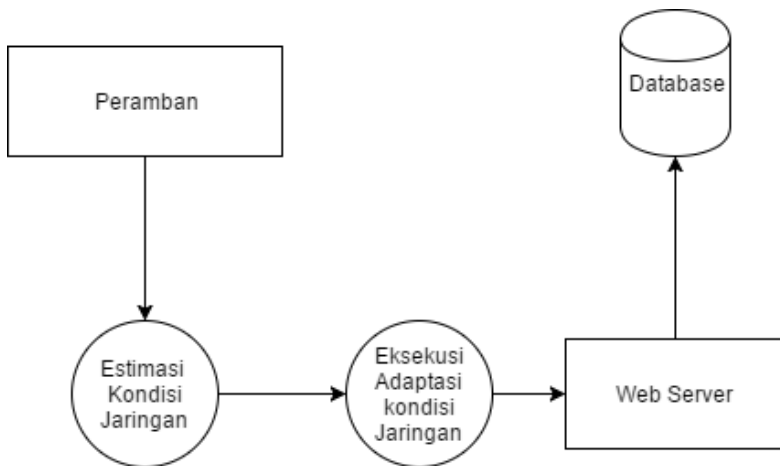
Seperti yang ditunjukkan pada Gambar 3.2, sistem ini akan menerima *HTTP request* dari peramban untuk membuka aplikasi. *Web server* mengirim konten yang diminta oleh peramban.. lalu data disimpan kedalam database.

3.3.2 Diagram Alir Data Level 0 Aplikasi

Diagram alir data level 0 ini merupakan dekomposisi dari proses utama pada diagram konteks. Diagram ini menggambarkan fungsionalitas yang terjadi pada proses di sistem ini. Diagram alir data level 0 ditunjukkan pada Gambar 3.3.

Diagram pada Gambar 3.3 menunjukkan proses-proses yang terjadi pada sistem. Proses awal yang terjadi yaitu peramban meminta *HTTP request* kepada *web server* untuk membuka aplikasi. *Web server* mengirimkan konten yang diminta oleh peramban. .

Setelah klien dapat mengirim dan menerima data, dilakukan estimasi kondisi jaringan. Estimasi kondisi jaringan dilakukan dengan cara mengecek kecepatan unggah yang dimiliki oleh klien. *Server* mengirim perintah pada klien untuk mengirim sebuah *file* untuk mengecek kecepatan yang dimiliki oleh klien .Estimasi yang telah dilakukan menghasilkan keluaran berupa kecepatan yang dimiliki oleh klien. Jadi pengiriman data selanjutnya dari klien ke *server* sesuai dengan kondisi jaringan yang ada.



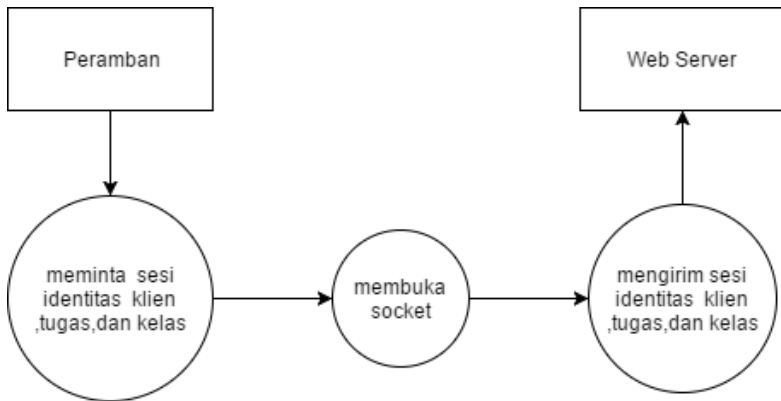
Gambar 3.3 Diagram Alir Data Level 0

3.4 Diagram Alir Aplikasi

Pada bagian ini akan dijelaskan secara lebih mendetail setiap proses yang terjadi pada sistem. Proses ini akan digambarkan menggunakan diagram alir. Hal ini dimaksudkan untuk mempermudah memahami alur kejar aplikasi pada tugas akhir ini. Diagram yang dibahas pada bagian ini adalah proses yang berjalan pada aplikasi diluar yang dikerjakan oleh pustaka pemrograman dan pustaka.

3.4.1 Diagram Alir Proses Inisialisasi Koneksi klien

Pada proses ini dilakukan inisialisasi koneksi klien yang bertujuan untuk menghubungkan antara peramban dengan *server* peer. Pada proses inisialisasi tersebut dibutuhkan identitas. klien satu dengan yang lain tidak boleh memiliki identitas yang sama karena identitas tersebut digunakan untuk keluar masuknya data pada *server*.

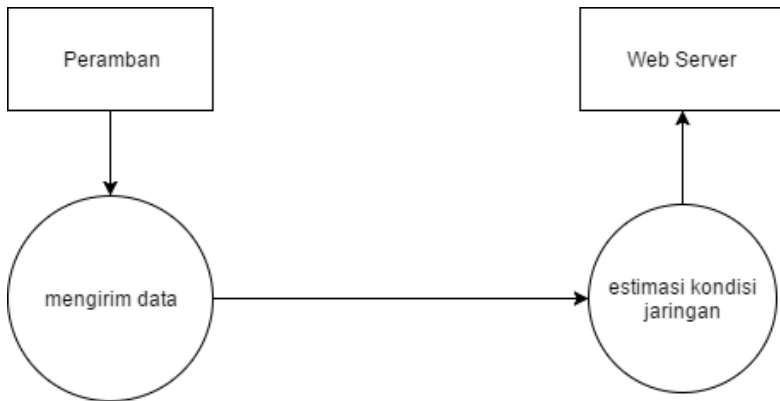


Gambar 3.4 Diagram Alir Koneksi Klien

Pada Gambar 3.4 terlihat bahwa peramban meminta identitas klien, kelas serta tugas yang digunakan untuk inisialisasi koneksi klien. Kemudian peramban membuat socket dengan *server*. Socket dimulai dengan mengirimkan identitas yang dimiliki oleh klien. *Server* akan menyimpan koneksi tersebut yang berfungsi agar dapat berhubungan dengan peer melalui identitas tersebut.

3.4.2 Diagram Alir Proses Estimasi Kondisi Jaringan

Pada proses ini dilakukan estimasi terhadap kondisi jaringan saat ini. Kondisi jaringan tersebut berupa kecepatan unggah klien.. Diagram alir proses ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Diagram Alir Estimasi Kondisi Jaringan

Dari Gambar 3.5 terlihat proses yang dilakukan dalam estimasi kondisi jaringan. Klien melakukan pengiriman data berupa *file* berukuran kecil. Setelah itu dilakukan estimasi kondisi jaringan untuk klien. *server* menerima data .

3.4.3 Diagram Alir Proses Eksekusi Adaptasi Uploading

Pada proses ini dilakukan proses eksekusi adaptasi Uploading. Adaptasi uploading tersebut berupa konstrain dari *file*. Konstrain tersebut berupa ukuran *Chunk file* yang akan dikirim. Diagram alir proses ini dapat dilihat pada Gambar 3.6.

Dari Gambar 3.6 terlihat proses yang dilakukan dalam eksekusi adaptasi uploading. Awalnya, data kondisi jaringan didapat dari estimasi kondisi jaringan klien dan *server*. Estimasi kondisi jaringan tersebut dijadikan sebagai argumen dari logika.



Gambar 3.6 Diagram Alir Eksekusi Adaptasi Uploading

3.5 Perancangan *Database*

Perancangan *database* diperlukan untuk menyimpan tugas klien. Pada Tugas Akhir ini dibuat dua tabel tambahan pada CMS yang ada yaitu tabel *kelas_tugas_pengumpulan* untuk menyimpan informasi *file* yang dimiliki klien dan tabel *kelas_tugas_pengumpulan_chunk* untuk menyimpan informasi dari potongan *file* yang disimpan saat proses pengunggahan.

3.5.1 Tabel *Kelas_tugas_pengumpulan*

| Kelas_tugas_pengumpulan | |
|-------------------------|-------------|
| id | int(11) |
| id_kelas | int(11) |
| id_materi | int(11) |
| id_tugas | int(11) |
| id_peserta_didik | int(11) |
| tgl_pengumpulan | Date |
| url_lampiran | text |
| id_status_tugas | int(11) |
| id_status_koreksi | int(11) |
| id_metadata | varchar(64) |
| total | int(11) |
| part | text |
| file_size | text |
| status_upload | int(11) |

Gambar 3.7 Tabel *Kelas_tugas_pengumpulan*

Pada Gambar 3.7 adalah tabel `Kelas_tugas_pengumpulan` yang digunakan untuk menyimpan identitas dan informasi dari proses pengumpulan tugas. Informasi yang disimpan adalah `id` untuk menyimpan nomor `id file` milik klien di dalam aplikasi, `id_kelas` untuk menyimpan nomor `id kelas` pada saat masuk ke dalam aplikasi, `id_materi` untuk menyimpan nomor `id materi tugas`, `id_peserta_didik` untuk menyimpan nomor `id peserta didik`, `tgl_pengumpulan` untuk menyimpan tanggal klien mengumpulkan , `url_lampiran` untuk menyimpan lokasi `file` di dalam `server`, `id_status` untuk menyimpan informasi status tugas, `id_status_koreksi` untuk informasi apakah tugas sudah dikoreksi atau belum, `id_metadata` untuk menyimpan nomor `id metadata file`, `total` untuk menyimpan jumlah potongan `file` berdasarkan skema adaptif, `part` untuk menyimpan ukuran potongan `file` berdasarkan skema adaptif, `file_size` untuk menyimpan ukuran `file` sebenarnya, `status_upload` untuk menyimpan informasi status upload.

3.5.2 Tabel kelas_tugas_pengumpulan_chunk

| Kelas_tugas_pengumpulan_chunk | |
|-------------------------------|-------------|
| id | int(11) |
| id_kelas | int(11) |
| id_pd | int(11) |
| id_metadata | varchar(64) |
| ori_name | text |
| file_name | text |
| num | int(11) |
| ukuran | int(11) |
| part | text |
| file_size | text |
| persen | int(11) |
| sukses | int(1) |

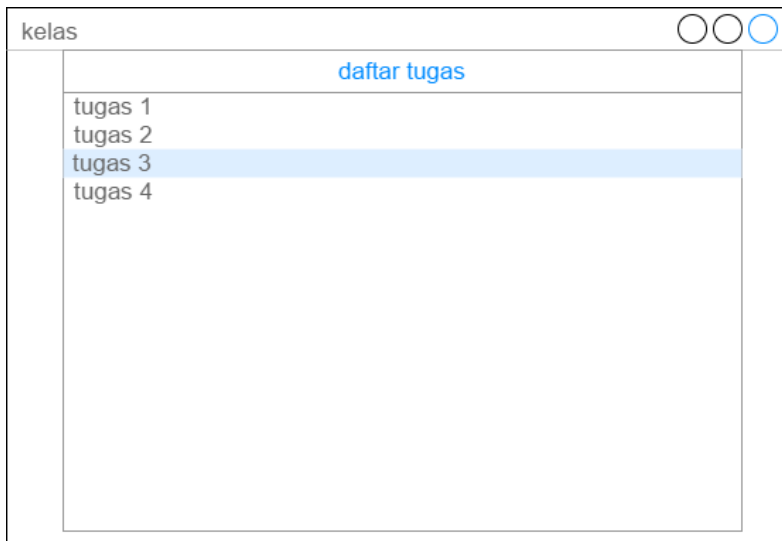
Gambar 3.8 Tabel kelas_tugas_pengumpulan_chunk

Pada Gambar 3.8 adalah tabel kelas_tugas_pengumpulan_chunk yang digunakan untuk menyimpan informasi dari kelas_tugas_pengumpulan_chunk. Informasi yang disimpan adalah id untuk menyimpan nomor id potongan file milik klien di dalam aplikasi, id_kelas untuk menyimpan nomor id kelas pada saat masuk ke dalam aplikasi, id_pd untuk menyimpan nomor id peserta didik, id_metadata untuk menyimpan nomor id metadata file, ori_name untuk menyimpan nama file saat diunggah, file_name untuk menyimpan nama file saat disimpan num untuk menyimpan nomor urutan file yang digunakan untuk menggabungkan file, ukuran untuk menyimpan jumlah file potongan, part untuk menyimpan ukuran potongan file, file_size untuk menyimpan ukuran file seutuhnya, persen untuk menyimpan persentase file yang telah berhasil diunggah, sukses untuk menandai file telah berhasil dikirim ke server.

3.6 Rancangan Antarmuka Sistem

3.6.1 Tampilan Tugas(semua /belum selesai/sudah selesai)

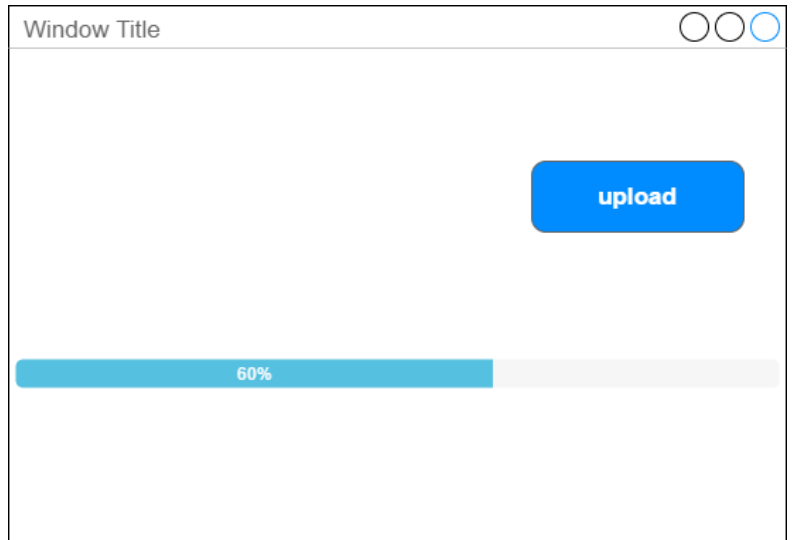
Pada Tugas Akhir ini, antarmuka sistem berupa tampilan web. Untuk itu diperlukan perancangan antarmuka yang untuk web ini. Rancangan desain web dapat dilihat pada Gambar 3.9 ,Gambar 3.10 dan Gambar 3.11.



Gambar 3.9 Desain Antarmuka Tampilan Web

3.6.2 Tampilan Upload Tugas

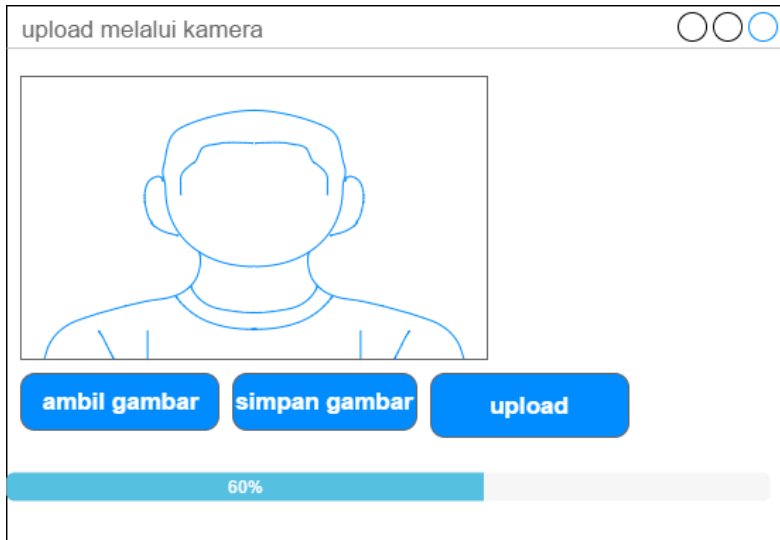
Pada Gambar 3.10 menunjukkan halaman web menampilkan halaman upload *file*. Di sisi atas merupakan tampilan dari fungsi mengumpulkan tugas melalui *file* ,di halaman ini klien dapat memilih *file* yang berada di perangkat klien untuk diunggah ke *server*.



Gambar 3.10 Desain Antarmuka Tampilan *upload file*

3.6.3 Tampilan Upload melalui Kamera

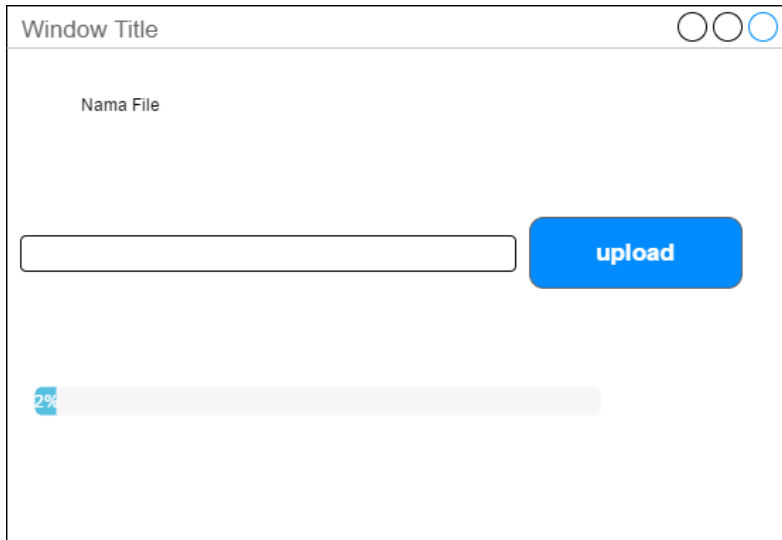
Pada Gambar 3.11 menunjukkan halaman web menampilkan halaman *upload file*. Di sisi atas merupakan tampilan dari fungsi mengumpulkan tugas melalui kamera yang terdapat pada perangkat klien ,di halaman ini klien dapat mengambil gambar melalui kamera yang berada di perangkat klien untuk diunggah ke *server*



Gambar 3.11 Desain Antarmuka Tampilan *upload melalui kamera*

3.6.4 Tampilan Melanjutkan Upload

Pada Gambar 3.12 menunjukkan halaman web menampilkan halaman melanjutkan upload. Apabila terjadi gangguan jaringan yang menyebabkan terputusnya proses upload, klien dapat melanjutkan proses upload dengan cara mengirimkan lagi *file* dengan nama dan ukuran yang sama.



Gambar 3.12 Desain Antarmuka Tampilan melanjutkan upload

3.6.5 Tampilan Daftar *File*

Pada Gambar 3.13 menunjukkan halaman web menampilkan halaman daftar *File*. Klien dapat melihat daftar *file* yang sudah pernah dikirim ke *server*. *File* yang terdapat di daftar tersebut dapat diunduh oleh klien.



Gambar 3.13 Desain Antarmuka Tampilan Daftar *File*

(Halaman ini sengaja dikosongkan)

BAB IV

IMPLEMENTASI PERANGKAT LUNAK

Setelah melewati proses analisis dan perancangan perangkat lunak, maka dilakukan implementasi sistem. Bab ini akan membahas implementasi dari perancangan sistem perangkat lunak yang telah dibahas pada bab sebelumnya.

4.1 Lingkungan Pembangunan Perangkat Lunak

Pembangunan perangkat lunak pada tugas akhir ini dibangun pada lingkungan yang akan dijabarkan pada bagian selanjutnya.

4.1.1 Lingkungan Perangkat Lunak

Pembangunan aplikasi tugas akhir ini menggunakan bantuan perangkat lunak sebagai berikut:

- Sistem Operasi Ubuntu 64 bit sebagai *server*.
- Sublime Text versi 2.0.2 sebagai IDE untuk pembangunan aplikasi web..
- MySQL versi 5.5 sebagai basis data aplikasi.

4.1.2 Lingkungan Perangkat Keras

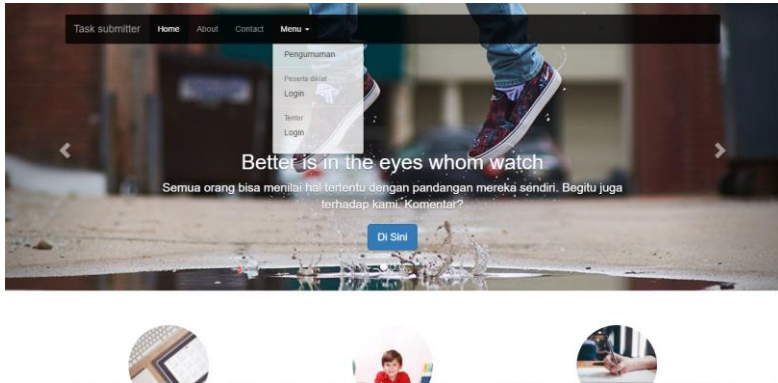
Spesifikasi perangkat keras yang digunakan untuk pembangunan aplikasi tugas akhir ini adalah menggunakan IBM Bladecenter sebagai *server* dari sistem. *Server* mempunyai spesifikasi 4GB RAM.

4.2 Mekanisme Perangkat Lunak

Pada subbab ini, akan dibahas implementasi dari *server* dan proses serta implementasi dari pengembangan plugin dalam tugas akhir ini.

4.2.1 Web Server

Web Server yang diimplementasikan menggunakan *server* Apache dengan dukungan PHP dan juga MySQL *server* yang akan digunakan oleh *elearning* sebagai sistem manajemen basis data. *Elearning* tersebut dapat diakses langsung melalui peramban. Gambar 4.2.1 merupakan halaman awal *Elearning*.



Gambar 4.1 Halaman Awal *Elearning*

4.2.2 Implementasi Fungsi

Pada Subbab-subbab berikut akan dijelaskan implementasi dari *uploadTugas* yang digunakan untuk menambahkan fungsi mengunggah *file* untuk tugas pada *Elearning*.

4.2.2.1 Fungsi Mengecek Kecepatan

Setelah murid memilih tugas yang ingin di unggah, murid akan masuk ke halaman *File Upload*. Pada awalnya *server* akan mengecek kecepatan unggah yang dimiliki oleh murid. Proses pengecekan dimulai dari *server* memberi perintah kepada peramban murid untuk mengirimkan sebuah data. Lalu saat *server*

menerima data tersebut ,*server* menghitung waktu jeda pengiriman data .Dengan proses tersebut didapatkan informasi kecepatan unggah yang akan digunakan untuk skema adaptif .Implementasi untuk fungsi mengecek kecepatan dapat dilihat di Gambar 4.2.

```
uploadSize = 1*1024*1024;

url = 'index.php';

myData = "d=";

    for(var i = 0 ; i < uploadSize ; i++)

        {

myData += "k";
        }

http.open("POST", url, true);

http.onreadystatechange = function()

{    if(http.readyState == 4 && http.status ==
200) {    endTime = (new Date()).getTime();
        showData();    }}

startTime = (new Date()).getTime();

http.send(myData);

var duration = (endTime - startTime) / 1000;

var kbps=((uploadSize/1024)
/duration).toFixed(2);

var mbps = ((uploadSize/1024/1024)/
duration).toFixed(2);
```

Gambar 4.2 Implementasi Mengecek kecepatan

4.2.2.2 Fungsi Adaptif Pemotongan File

Setelah informasi kecepatan unggah milik pengguna didapat, *server* melakukan skema adaptif dalam proses pengunggahan *file*. Skema adaptif untuk proses pengunggahan *file* dapat dijelaskan sebagai berikut:

- Apabila kecepatan unggah pengguna lebih dari 1MBps maka *file* akan dipecah menjadi 3Megabit untuk setiap bagian.
- Apabila kecepatan unggah pengguna 1Mbps maka *file* akan dipecah menjadi 1 Megabit untuk setiap bagian.
- Apabila kecepatan unggah pengguna 1MBps sampai 512Kbps maka *file* akan dipecah menjadi 512 Kilobit untuk setiap bagian
- Apabila kecepatan lebih rendah daripada itu maka *file* akan dipecah menjadi 256 Kilobit untuk setiap bagian

Implementasi untuk fungsi mengecek kecepatan dapat dilihat digambar 4.3

```
var file = $('#uploadFile')[0].files[0];
var lamaWaktu = parseInt($("#speedTest").text());
if( lamaWaktu <= 1 ){
    var chunkSize = 1024 * 3000;
} else if( lamaWaktu > 1 && lamaWaktu <= 3)
{ var chunkSize = 1024 * 1000;
} else if( lamaWaktu > 3 && lamaWaktu <= 5)
{ var chunkSize = 512 * 1000;
} else{ var chunkSize = 256 * 1000;
}
```

Gambar 4.3 Implementasi Fungsi Adaptif pemotongan File

4.2.2.3 Fungsi Pengiriman *File*

Setelah perhitungan pemotongan *file*, *file* siap untuk dikirimkan ke *server*. *File* yang dikirimkan menggunakan *Json*, dikirimkan per bagian. Variabel yang dikirimkan yaitu urutan, total, token, chunksize dan *Filesize*. Apabila *file* telah terkirim semua *file* akan di gabungkan kembali menggunakan *merge.php*. Implementasi untuk fungsi pengiriman *file* dapat dilihat digambar 4.4 dan 4.5

```
var respon = xhr.responseText;

var jsonParse = JSON.parse(respon);

if(jsonParse.result=="sukses"){

    var prosenRespon =
parseFloat(jsonParse.prosentase);

    var id= jsonParse.id;

    var totalPros =
parseFloat($("#totalPros").text());

    var persen = prosenRespon+totalPros;

    $('#progressFile').progress({ value: urutan,
total: total});
    $("#totalPros").text(persen);

    var persenBulat = Math.round(persen);

    if(urutan == total){

        $(".resultAll").show().html("<i>Waiting    for
combining "+total+" parts files...</i><br/>");
        var namaFile = file.name;
```

Gambar 4.4 Implementasi Fungsi Pengiriman *File*(1)

```

        $.post('upload/merge.php',{nama: namaFile,
totalChunk: total, metadata: id},
        function(hasil){
            $(".resultAll").append(hasil);
            $("#uploadFilesButton").removeClass('loading');

    }else{ urutan++;

            var chunk = urutan - 1;

            var offset = chunk*chunkSize;

            uploadServer(file,file.slice(offset,offset+
chunkSize),urutan,total,token,chunkSize,
fileSize)}

```

Gambar 4.5 Implementasi Fungsi Pengiriman *File*(2)

4.2.2.4 Fungsi Melanjutkan Pengiriman *File*

Apabila jaringan terputus,saat proses pengiriman *file* oleh pengguna ,maka pengguna dapat melanjutkan proses pengiriman *file*. Pengguna dapat melanjutkan proses pengiriman dengan cara mengirimkan *file* yang sama(dengan syarat nama dan ukuran sama seperti *file* yang gagal terkirim . Implementasi untuk fungsi melanjutkan pengiriman *file* dapat dilihat digambar 4.6.

```

var file = $('#uploadFile')[0].files[0];
var chunkSize = <?php echo $part; ?>;
var fileSize = file.size;
var fileName = file.name;
var chunks = Math.ceil(file.size/chunkSize);
var chunk = nmArray[arrayStart] - 1;
var total      = totalArray-1;
var token = "<?php echo $metadata; ?>";

    var urutan  = chunk+1;

    var offset  = chunk*chunkSize;

if( fileSize == <?php echo $fileSize ?> &&
   fileName == "<?php echo $fileName ?>" ){
    uploadServer(file,file.slice(offset,offset+chunkSize),urutan,total,token,chunkSize,fileSize);
}else{

alert("Silahkan memasukkan File yang sama nama
dan ukurannya!");

}

```

Gambar 4.6 Implementasi fungsi melanjutkan pengiriman *File*

4.2.2.5 Fungsi Mengambil Gambar melalui Kamera

Aplikasi ini juga memberikan cara alternatif untuk mengumpulkan tugas,yaitu dengan cara mengambil photo dari tugas tersebut melalui kamera perangkat yang dimiliki pengguna.Implementasi fungsi Mengambil gambar melalui kamera dapat dilihat pada Gambar 4.7.

```

Webcam.snap( function(data_uri) {

    document.getElementById('results').innerHTML +=

    '';

    $("#simpanGambar").attr("href", data_uri);

    window.dataImg = data_uri;

} );

```

Gambar 4.7 Implementasi fungsi mengambil gambar dari kamera

4.2.2.6 Fungsi Mengirim *File* melalui Kamera

Fungsi pengiriman *File* melalui kamera memiliki sedikit perbedaan dengan fungsi mengirim *file* biasa. Pada fungsi ini *file* gambar akan dipecah sebesar 100 Kilobit lalu dikirimkan perbagian .Setelah bagian-bagian dari gambar sudah terkirim ,lalu *file* gambar akan digabungkan dan tersimpan di *server*. Pada proses mengirim *File* melalui kamera tidak terdapat proses adaptif ,tapi gambar yang diambil melalui perangkat dapat disimpan dan dikirim melaui proses pengiriman *file* biasa. Implementasi fungsi mengirim gambar melalui kamera dapat dilihat pada Gambar 4.8.

```

var file = dataURIToBlob(dataImg);

$("#reportProgress").hide();

$(".resultAll").hide();

$("#uploadFilesButton").addClass('loading');

$("#totalPros").text('0');

var chunkSize = 1024 * 100;

var fileSize = file.size;

var chunks = Math.ceil(file.size/chunkSize);

var chunk = 0;

console.log('file size..',fileSize);

$.post('get-token.php',{total: chunks,perPart:
chunkSize, fileSize: file.size},
function(tokenResult){

    var token = tokenResult;

    var urutan = chunk+1;

    var offset = chunk*chunkSize;

    uploadServer(file,file.slice(offset,offset+chunkSize),urutan,chunks,token,chunkSize,fileSize);

$("#reportProgress").show(500);

});

```

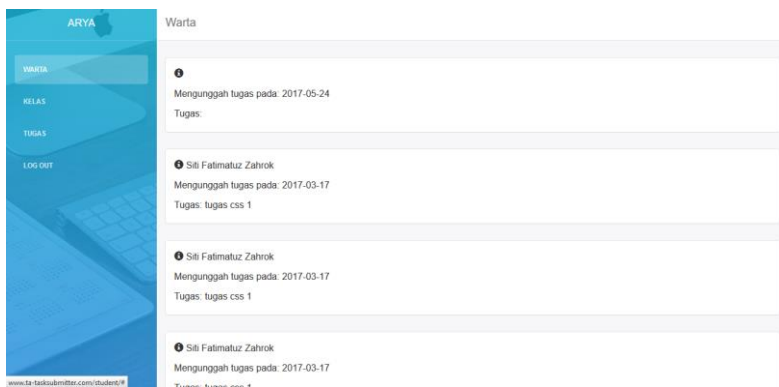
Gambar 4.8 Implementasi fungsi mengirim *file* dari kamera

4.3 Implementasi Antarmuka Perangkat Lunak

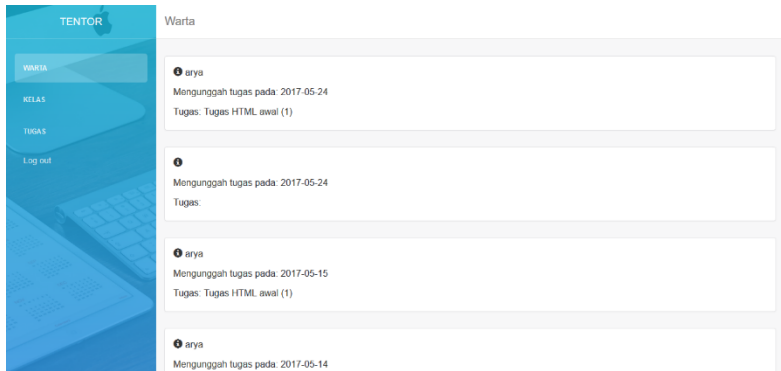
Sesuai dengan rancangan antarmuka yang telah dibuat pada BAB III, maka implementasi antarmuka perangkat lunak akan dijelaskan pada subbab-subbab berikut.

4.3.1 Tampilan Awal

Pada tampilan awal, jenis hak akses yang berbeda akan mempengaruhi tampilan menu navigasi. Selain itu, ini memperjelas *identity* termasuk *user role* dari pengguna yang sedang masuk yang bisa dilihat pada bagian paling atas dari Gambar 4.9 dan Gambar 4.10.



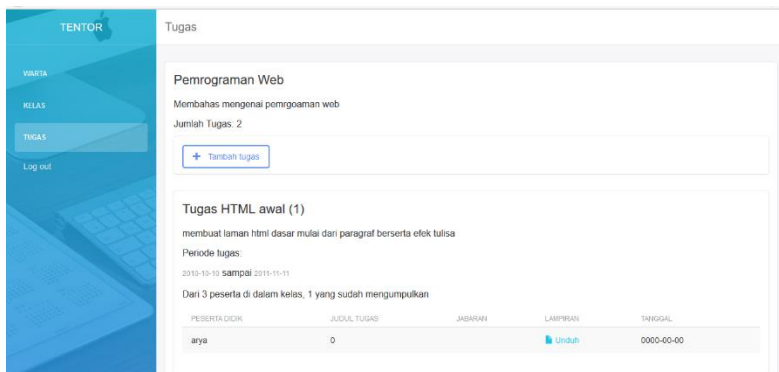
Gambar 4.9 Tampilan awal murid



Gambar 4.10 Tampilan awal guru

4.3.2 Tampilan tugas guru

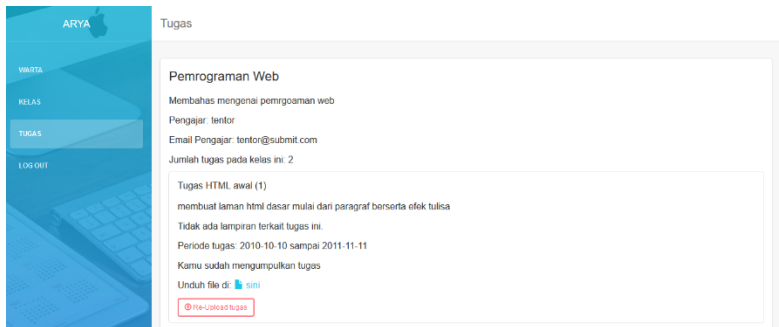
Pada tampilan ini, guru dapat melihat semua tugas yang ada dari semua kelas yang dimiliki, melihat semua tugas yang belum selesai dan telah selesai dari semua kelas, dan melihat semua tugas dari masing-masing kelas dalam sistem. tampilan ini dapat dilihat pada Gambar 4.11



Gambar 4.11 Tampilan awal tugas guru

4.3.3 Tampilan awal tugas murid

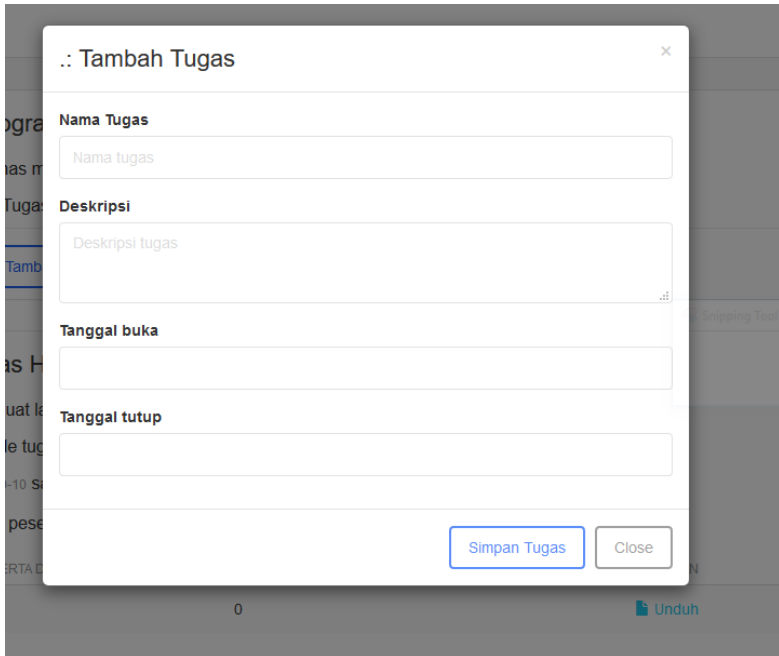
Pada tampilan ini, murid dapat melihat semua tugas yang ada dari semua kelas yang dimiliki, melihat semua tugas yang belum selesai. tampilan ini dapat dilihat pada Gambar 4.12



Gambar 4.12 Tampilan awal tugas murid

4.3.4 Tampilan Tambah Tugas

Pada tampilan ini, guru dapat menambahkan tugas kepada murid yang berada dikelasnya. Tugas yang ditambahkan dari halaman ini akan dapat dilihat oleh murid dari tampilan melihat tugas kelas masing-masing. Kolom yang diwajibkan untuk diisi dan dipilih adalah nama tugas, deskripsi, tanggal buka, dan tanggal tutup. Setelah menekan tombol Save, informasi dari tugas yang dimasukkan dari tampilan ini akan disimpan ke dalam basis data. Gambar 4.13 merupakan implementasi antarmuka tambah tugas



..: Tambah Tugas

Nama Tugas

Deskripsi

Tanggal buka

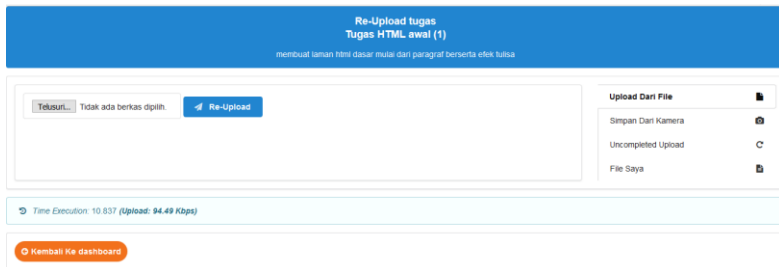
Tanggal tutup

Simpan Tugas Close

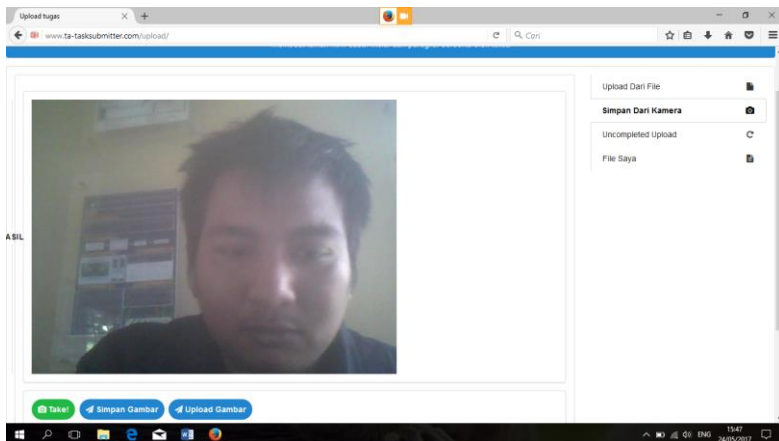
Gambar 4.13 Tampilan tambah tugas

4.3.5 Tampilan Upload Tugas

Pada tampilan ini, murid dapat mengirim tugas yang diberikan guru yang berada dikelasnya. Murid dapat memilih mengirim tugas melalui *file* maupun mengambil gambar melalui kamera yang terdapat pada perangkat yang digunakan. Gambar 4.14 merupakan implementasi Upload tugas melalui *file*, sedangkan Gambar 4.15 melalui kamera.



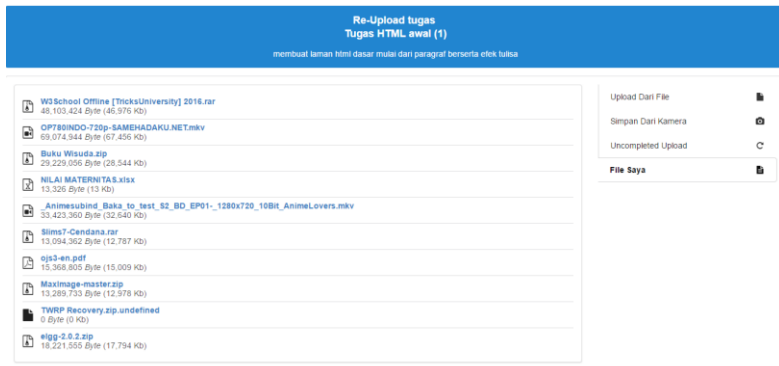
Gambar 4.14 Tampilan Upload Tugas melalui *File*



Gambar 4.15 Tampilan Upload Tugas melalui Kamera

4.3.6 Tampilan Daftar *File*

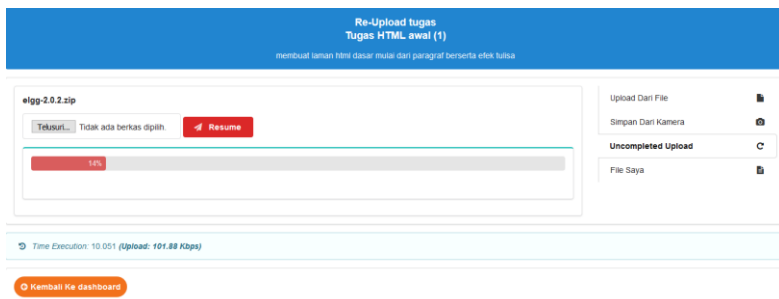
Pada tampilan ini, pengguna dapat melihat daftar *file* milik pengguna yang sudah pernah dikirim. Dengan daftar ini juga pengguna dapat mengunduh kembali *file-file* yang telah terkirim ke *server*. Gambar 4.16 merupakan implementasi tampilan Daftar *File*.



Gambar 4.16 Tampilan Daftar *File*

4.3.7 Tampilan Lanjut Upload

Pada tampilan ini, apabila terjadi gangguan saat proses pengiriman *file*. Maka murid dapat melanjutkan proses pengiriman *file* dengan cara mengirimkan *file* yang sama melalui tampilan ini. Gambar 4.17 merupakan Implementasi tampilan Lanjut Upload.



Gambar 4.17 Tampilan Lanjut Upload

(Halaman ini sengaja dikosongkan)

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dibahas mengenai hasil ujicoba fungsionalitas dan performa dari aplikasi tugas akhir. Tahapan ujicoba ini akan dilakukan dalam beberapa skenario yang akan dibahas pada bagian selanjutnya dari bab ini.

5.1 Lingkungan Uji Coba

Dalam melakukan uji coba aplikasi tugas akhir ini, dilakukan menggunakan 2 komputer yang masing-masing spesifikasi nya sebagai berikut.

Spesifikasi Perangkat Komputer 1:

- Intel Core i5
- Windows 10 64-bit sebagai Sistem Operasi
- 4GB RAM
- Mozilla Firefox

Spesifikasi Perangkat Komputer 2:

- Intel Core i5
- Windows 10 64-bit sebagai Sistem Operasi
- 4GB RAM
- Google Chrome versi 43

Spesifikasi *Server*

- IBM blade *server*
- Ubuntu versi 12
- 4GB RAM

5.2 Data Uji Coba

Data uji coba yang digunakan adalah beberapa jenis *file* dengan ukuran yang berbeda. *File* yang dikirim sebagai uji coba yaitu rar dengan ukuran 17.6 Megabyte, gambar berukuran 500 Kilobyte serta gambar didapat melalui kamera perangkat.

5.3 Skenario Uji Coba

Proses uji coba mengenai aplikasi ini akan dibagi ke dalam beberapa skenario. Uji coba bertujuan dimaksudkan untuk menguji fungsionalitas serta performa dari aplikasi tugas akhir ini. Pengujian fungsionalitas bertujuan untuk memastikan aplikasi berjalan sesuai fungsi yang diharapkan. Uji coba dilakukan dengan menguji kualitas upload pada *elearning*. Uji coba Performa akan dilakukan pada 3 kondisi yaitu pada kondisi normal, Regular 3g dan GPRS menggunakan fitur yang tersedia pada peramban Google Chrome. Selain itu, akan dibandingkan metode pengunggahan pada tugas akhir ini dengan metode pengunggahan biasa. Keterangan keadaan jaringan tersebut dapat dilihat pada tabel berikut:

Tabel 5.1 Skenario Kondisi Jaringan

| Kondisi | Batas Kecepatan Unduh(kbps) | Batas Kecepatan Unggah (kbps) | Latensi (ms) |
|---------|-----------------------------|-------------------------------|--------------|
| Normal | - | - | - |
| 3g | 750 | 250 | 100 |
| GPRS | 50 | 20 | 500 |

5.3.1 Uji Coba Fungsionalitas

Uji coba fungsionalitas merupakan sebuah pengujian yang dilakukan terhadap jalannya fungsi-fungsi utama pada sistem yang telah dibuat. Pengujian dilakukan ke seluruh fungsi sistem. Uji coba fungsionalitas ini meliputi semua alur program yang sudah dijelaskan pada bab sebelumnya diantaranya sebagai berikut.

5.3.1.1 Uji Coba Mengecek Kecepatan

Uji coba ini dilakukan dengan cara murid menggunakan Komputer 1 melakukan upload tugas yang akan mengarahkannya ke halaman upload tugas. *Server* akan memberikan perintah kepada peramban yang digunakan untuk mengupload *file* berupa teks . Uji coba dilakukan dengan scenario 5.1 seperti yang tertera pada Tabel 5.2.

Tabel 5.2 Prosedur Uji Coba Mengecek Kecepatan

| | |
|---------------------------|--|
| ID | UJ-01 |
| Referensi Use Case | UC-01 |
| Nama | Uji Coba Mengecek Kecepatan |
| Tujuan Uji Coba | Menguji fitur untuk menampilkan kecepatan pengguna. |
| Kondisi Awal | Pengguna berada pada antarmuka Upload Tugas. |
| Skenario 1 | Aplikasi ingin mengetahui kecepatan unggah pengguna |

| | |
|-----------------------|---|
| Masukan | - |
| Keluaran | Aplikasi menampilkan kecepatan unggah pengguna. |
| Hasil Uji Coba | Berhasil |

Upload tugas

Telusuri...

elgg-2.0.2.zip

Upload

Upload Dari File

Simpan Dari Kamera

Uncompleted Upload

File Saya

Time Execution: 10.935 (Upload: 93.64 Kbps)

Kembali Ke dashboard

Gambar 5.1 Hasil Uji Coba Mengecek Kecepatan

Pada Gambar 5.1 dapat terlihat kecepatan yang dimiliki pengguna. Informasi ini nantinya akan digunakan sebagai ukuran untuk melakukan strategi pengiriman *file* yang adaptif terhadap keadaan jaringan. Proses ini dimulai dengan *server* memerintahkan peramban pengguna untuk mengirimkan *file* text berukuran 1

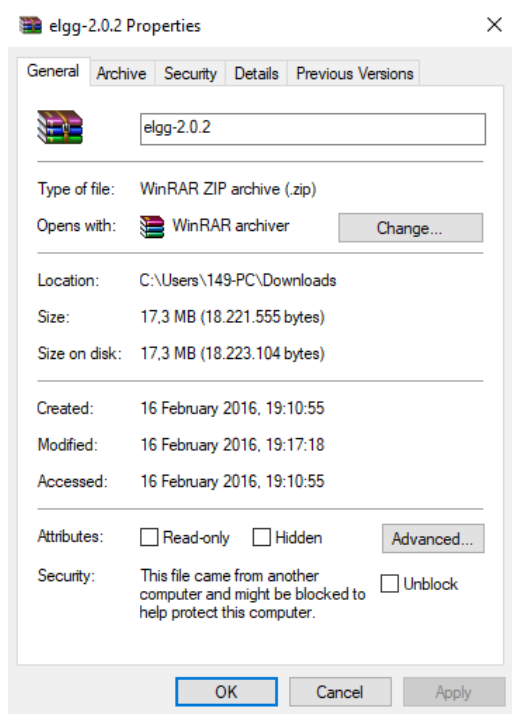
Megabyte seperti yang terlihat di *console* peramban yang dimiliki pengguna pada Gambar 5.2 .

```
start: 1495593304084
end :1495593315019
durasi: 10.935
size: 1048576
=====
Speed Result: 93.64 /Kbps
Speed Result: 0.09 /Mbps
=====
```

**Gambar 5.2 Hasil Uji Coba Mengecek Kecepatan melalui
*Console***

5.3.1.2 Uji Coba Pengiriman *File* 1

Uji coba ini dilakukan dengan murid menggunakan Komputer 1 melakukan upload tugas ke *server* yang akan mengarahkannya ke halaman upload tugas. *File* yang digunakan pada ujicoba ini berupa *file* bertipe rar yang berukuran 17.3 Megabyte .Informasi lengkap tentang *file* ini dapat terlihat pada Gambar 5.3. Uji coba dilakukan dengan skenario seperti yang tertera pada Tabel 5.3.

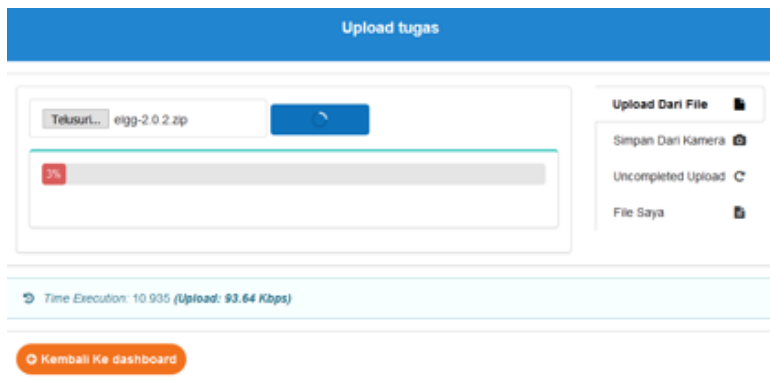


Gambar 5.3 Informasi *file* bertipe rar

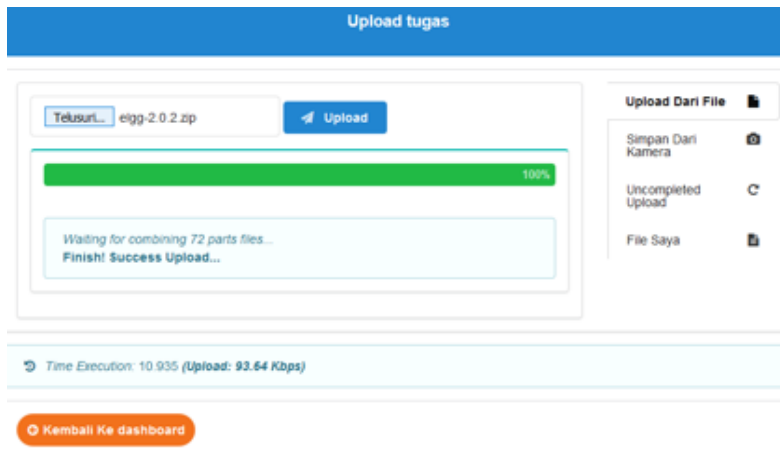
Tabel 5.3 Prosedur Uji Coba Pengiriman *File* 1

| | |
|---------------------------|--|
| ID | UJ-02 |
| Referensi Use Case | UC-02 |
| Nama | Uji Coba Pengiriman <i>File</i> 1 |

| | |
|------------------------|---|
| Tujuan Uji Coba | Menguji fitur untuk pengiriman <i>file</i> . |
| Kondisi Awal | Pengguna berada pada antarmuka Upload Tugas. |
| Skenario 1 | Pengguna mengirimkan <i>file</i> tugas |
| Masukan | <i>File</i> tugas bertipe rar berukuran 17.3 Megabyte |
| Keluaran | <i>File</i> tugas terkirim. |
| Hasil Uji Coba | Berhasil |



Gambar 5.4 Proses pengiriman *File* pada Ujicoba Pengiriman *File* 1



Gambar 5.5 Proses pengiriman *File* Selesai pada Ujicoba Pengiriman *File* 1

Pada Gambar 5.4 proses pengiriman *file* ini ,murid memiliki kecepatan unggah *file* sebesar 93.Kilobit perdetik . Menurut Bab 4.2.2 dengan kecepatan unggah yang dimiliki murid, *file* tugas milik murid yang dikirim akan dipecah menjadi ukuran 17,6 Megabyte untuk setiap potongannya dan akan dikirim per bagian sampai semua bagian terkirim.Karena ukuran *file* sebesar 17.3 Megabyte maka *file* akan dipotong menjadi 72 bagian.

Setelah semua bagian terkirim *file* akan diolah oleh *server* untuk digabungkan kembali dan disimpan pada *server* seperti yang terlihat pada Gambar 5.5. Informasi ukuran potongan *file* dapat terlihat pada Gambar 5.6 yang merupakan tampilan *console* dari jaringan pada peramban yang digunakan oleh murid.

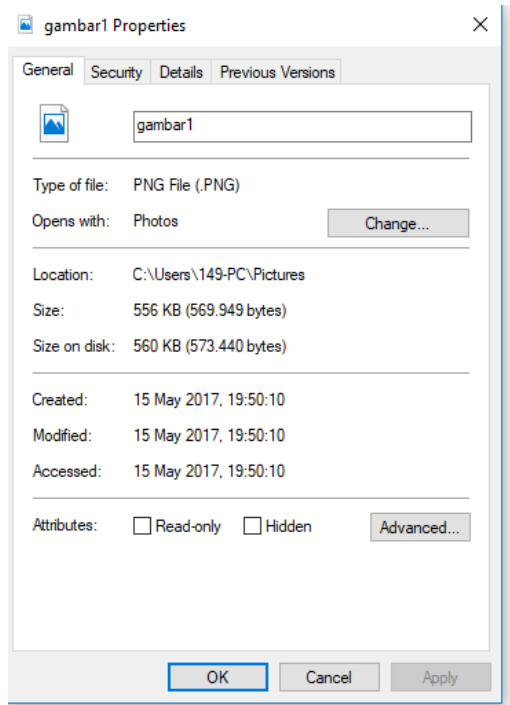
▼ Header permintaan

```
Accept */*
Accept-Encoding gzip, deflate
Accept-Language id,en-US;q=0.7,en;q=0.3
Connection keep-alive
Content-Length 256655
Content-Type multipart/form-data; boundary=-----275422436217287
Cookie __cfduid=d03bc7116d34df014c32eade4d51e8c901495586840; PHPSESSID=om05tt9405vtedddr749ho4rk0
Host www.ta-tasksubmitter.com
Referer http://www.ta-tasksubmitter.com/upload/
User-Agent Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101 Firefox/53.0
```

Gambar 5.6 Hasil Uji Coba Mengirim *File 1* melalui *Console*

5.3.1.3 Uji Coba Pengiriman tugas melalui *File 2*

Uji coba ini dilakukan dengan murid menggunakan Komputer 2 melakukan upload tugas ke *server* yang akan mengarahkannya ke halaman upload tugas. *File* yang digunakan pada ujicoba ini berupa *file* bertipe PNG yang berukuran 566 Kilobyte. Informasi lengkap tentang *file* ini dapat terlihat pada Gambar 5.7. Uji coba dilakukan dengan skenario seperti yang tertera pada Tabel 5.4.

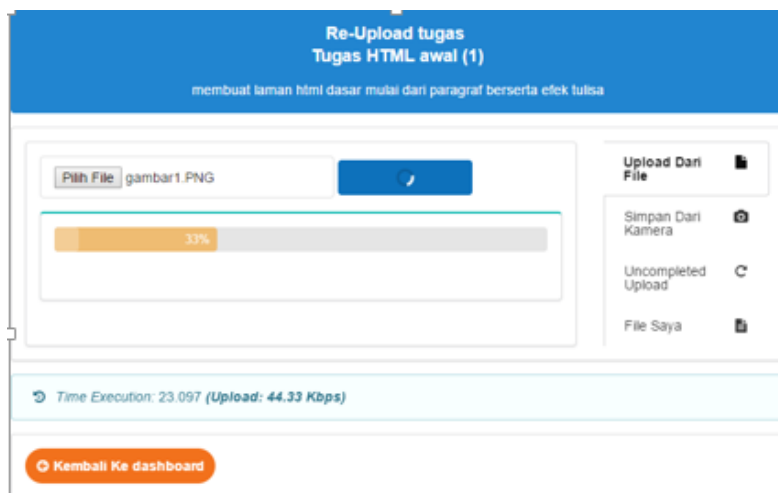


Gambar 5.7 Informasi *file* bertipe PNG

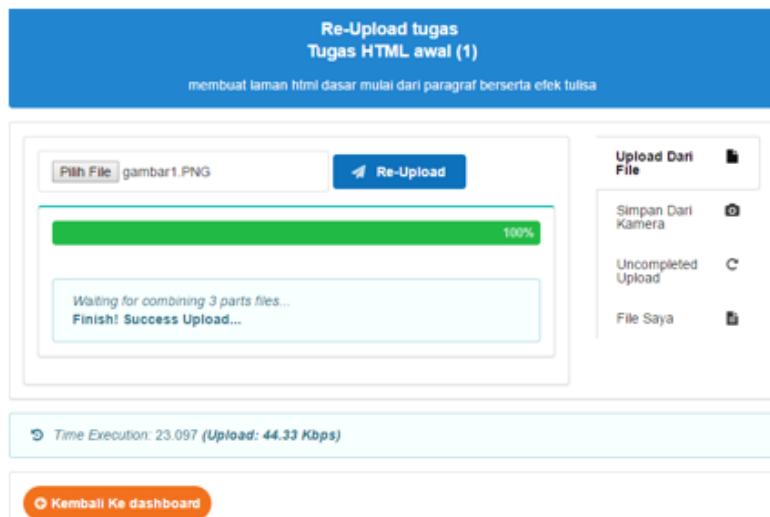
Tabel 5.4 Prosedur Uji Coba Pengiriman tugas melalui *File 2*

| | |
|---------------------------|--|
| ID | UJ-03 |
| Referensi Use Case | UC-03 |
| Nama | Uji Coba Pengiriman <i>File 2</i> |

| | |
|------------------------|---|
| Tujuan Uji Coba | Menguji fitur untuk pengiriman <i>file</i> . |
| Kondisi Awal | Pengguna berada pada antarmuka Upload Tugas. |
| Skenario 1 | Pengguna mengirimkan <i>file</i> tugas |
| Masukan | <i>File</i> tugas bertipe PNG berukuran 566 Kilobit |
| Keluaran | <i>File</i> tugas terkirim. |
| Hasil Uji Coba | Berhasil |



Gambar 5.8 Proses pengiriman tugas melalui *File* pada Ujicoba Pengiriman *File* 2



Gambar 5.9 Proses pengiriman tugas melalui *File* Selesai pada Ujicoba Pengiriman *File* 2

Pada Gambar 5.8 proses pengiriman *file* ini ,murid memiliki kecepatan unggah *file* sebesar 44.Kilobit perdetik . Menurut Bab 4.2.2 dengan kecepatan unggah yang dimiliki murid, *file* tugas milik murid yang dikirim akan dipecah menjadi ukuran 256 Kilobit untuk setiap potonganny dan akan dikirim per bagian sampai semua bagian terkirim.Karena ukuran *file* sebesar 566 Kilobyte maka *file* akan dipotong menjadi 3 bagian.

Setelah semua bagian terkirim *file* akan diolah oleh *server* untuk digabungkan kembali dan disimpan pada *server* seperti yang terlihat pada Gambar 5.9. Informasi ukuran potongan *file* dapat terlihat pada Gambar 5.10 yang merupakan tampilan *console* dari jaringan pada peramban yang digunakan oleh murid.

| | | | |
|--------------|--|--------|-----------|
| 14:55:04.265 | ::ChunkInfo:: | 256000 | VM289:156 |
| 14:55:04.311 | Total: | 3 | VM289:164 |
| 14:55:05.876 | ►XHR finished loading: POST "http://www.ta-tasksubmitter.com/upload/get-token.php". | | |
| 14:55:11.157 | ►XHR finished loading: POST "http://www.ta-tasksubmitter.com/upload/upload/index.php". | | |
| 14:55:16.497 | ►XHR finished loading: POST "http://www.ta-tasksubmitter.com/upload/upload/index.php". | | |
| 14:55:18.105 | ►XHR finished loading: POST "http://www.ta-tasksubmitter.com/upload/upload/index.php". | | |
| 14:55:18.787 | ►XHR finished loading: POST "http://www.ta-tasksubmitter.com/upload/upload/merge.php". | | |

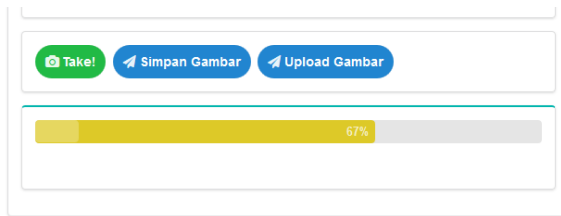
Gambar 5.10 Hasil Uji Coba Mengirim tugas melalui *File* 2 melalui *Console*

5.3.1.4 Uji Coba Pengiriman tugas Melalui Kamera

Uji coba ini dilakukan dengan murid menggunakan Komputer 1 melakukan upload tugas yang akan mengarahkannya ke halaman upload tugas. Murid akan melakukan pengumpulan tugas menggunakan kamera yang berada pada perangkat yang dimiliki . Uji coba dilakukan dengan skenario seperti yang tertera pada Tabel 5.5

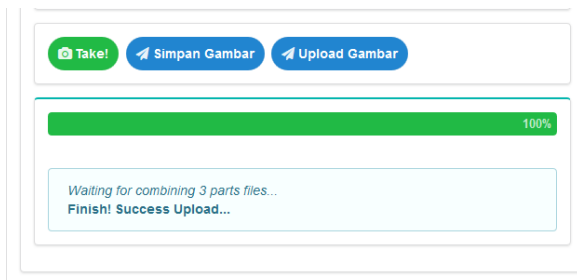
Tabel 5.5 Prosedur Uji Coba Pengiriman Tugas Melalui Kamera

| | |
|---------------------------|---|
| ID | UJ-05 |
| Referensi Use Case | UC-05 |
| Nama | Uji Coba Pengiriman Tugas Melalui kamera |
| Tujuan Uji Coba | Menguji fitur untuk pengiriman Tugas melalui kamera . |
| Kondisi Awal | Pengguna berada pada antarmuka Upload Tugas. |
| Skenario 1 | Pengguna mengirimkan tugas melalui kamera |
| Masukan | <i>File</i> gambar yang diambil melalui kamera perangkat yang dimiliki pengguna |
| Keluaran | <i>File</i> tugas terkirim. |
| Hasil Uji Coba | Berhasil |



🕒 Time Execution: 18.656 (Upload: 54.89 Kbps)

Gambar 5.11 Hasil Uji Coba Proses Mengirim Gambar melalui Kamera Perangkat



🕒 Time Execution: 18.656 (Upload: 54.89 Kbps)

Gambar 5.12 Hasil Uji Coba Proses Selesai Mengirim Gambar melalui Kamera Perangkat

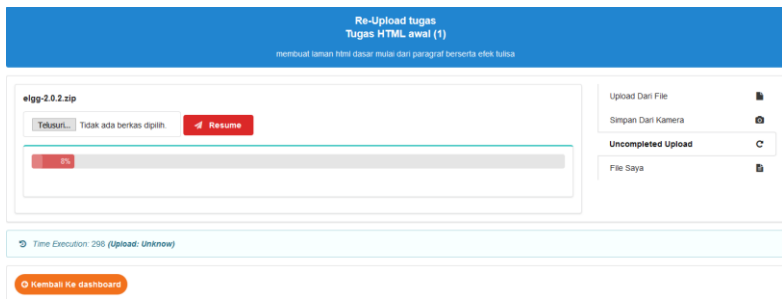
Pada Gambar 5.15 proses pengiriman *file* ini ,murid memiliki kecepatan unggah *file* sebesar 54.Kilobit perdetik . Menurut Bab 4.2.6, *file* tugas milik murid yang dikirm akan dipecah menjadi ukuran 100 Kilobit untuk setiap potonganny dan akan dikirim per bagian sampai semua bagian terkirim.

5.3.1.5 Uji Coba Lanjut Upload

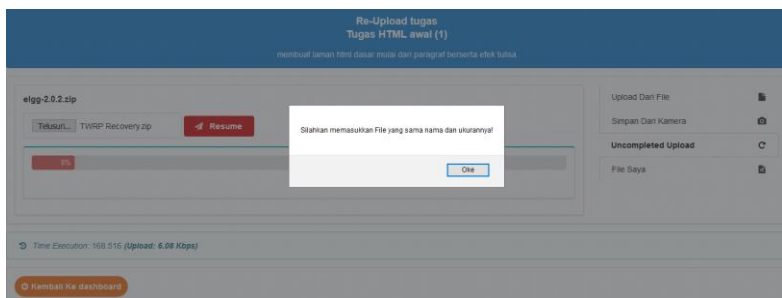
Uji coba ini dilakukan dengan murid menggunakan Komputer 1 mengalami gangguan saat mengirimkan tugas. Lalu murid melanjutkan proses pengiriman *file* yang belum terkirim. . Uji coba dilakukan dengan skenario seperti yang tertera pada Tabel 5.6

Tabel 5.6 Prosedur Uji Coba Melanjutkan Pengiriman Tugas

| | |
|---------------------------|---|
| ID | UJ-06 |
| Referensi Use Case | UC-05 |
| Nama | Uji Coba Melanjutkan Pengiriman Tugas |
| Tujuan Uji Coba | Menguji fitur untuk melanjutkan pengiriman tugas . |
| Kondisi Awal | Pengguna berada pada antarmuka Upload Tugas. |
| Skenario 1 | Pengguna melanjutkan pengiriman tugas |
| Masukan | Pengguna mengirimkan <i>file</i> tugas yang gagal terkirim sebelumnya |
| Keluaran | <i>File</i> tugas terkirim. |
| Hasil Uji Coba | Berhasil |



Gambar 5.13 *File yang belum selesai terkirim*



Gambar 5.14 *File yang Dipilih Tidak Sama dengan File yang Diminta*

Pada Gambar 5.17 terlihat bahwa murid memiliki 1 tugas yang gagal terkirim. Maka murid melakukan pengiriman ulang untuk melanjutkan pengiriman tugas yang terhenti. *File* tugas yang dikirimkan harus memiliki nama *file* dan ukuran yang sama apabila tidak maka akan muncul peringatan seperti yang diperlihatkan gambar 5.18. Setelah proses pengiriman ulang selesai maka *file* tugas akan digabungkan oleh *server* dan disimpan pada *server* seperti yang terlihat pada gambar 5.19.

Kecepatan yang ditempuh selama ujicoba serta jumlah potongan *file* yang dibuat dapat dilihat dari tabel 5.7.

5.3.2 Uji Coba Performa

Pada bagian ini dilakukan uji coba performa untuk mengetahui perilaku dari sistem ketika dijalankan pada keadaan sebenarnya. Uji coba ini dilakukan menggunakan skenario perubahan kondisi jaringan dengan cara menggunakan fitur penyesuaian kondisi jaringan yang terdapat pada Google Chrome. Penyesuaian kondisi jaringan pada Google Chrome memungkinkan menguji situs pada berbagai koneksi jaringan, termasuk Edge, 3G, dan bahkan offline. Ini juga membatasi throughput pengunduhan dan pengunggahan maksimum. Manipulasi latensi memaksa penundaan minimum pada round-trip time (RTT) koneksi. Pada uji coba ini akan dilakukan beberapa pengiriman *file* yang sama berukuran 1MB dalam keadaan kondisi jaringan yang berbeda.

Tabel 5.7 Kecepatan Upload pada saat Uji Coba

| No | Keadaan jaringan | Kecepatan upload(Kbps) |
|----|------------------|------------------------|
| 1 | Normal | 55.18 |
| 2 | Regular 3g | 23.29 |
| 3 | GPRS | 2.44 |

Tabel 5.8 Hasil Uji Coba Performa dalam keadaan normal

| No | Waktu Pengiriman <i>Server</i> (Jam:menit:detik) | Waktu Penerimaan <i>Server</i> (Jam:menit:detik) | Waktu Selisih(detik) |
|-----------|--|--|----------------------|
| 1 | 20:15:30 | 20:16:03 | 33 |
| 2 | 20:16:46 | 20:17:08 | 22 |
| 3 | 20:18:17 | 20:18:48 | 31 |
| 4 | 20:20:04 | 20:20:35 | 31 |
| 5 | 20:21:22 | 20:21:56 | 34 |
| 6 | 20:22:38 | 20:23:12 | 34 |
| 7 | 20:24:05 | 20:24:26 | 21 |
| 8 | 20:25:09 | 20:25:39 | 30 |
| 9 | 20:26:16 | 20:26:42 | 26 |
| 10 | 20:28:03 | 20:28:32 | 29 |
| Rata-rata | | | =29.1 |

**Tabel 5.9 Hasil Uji Coba Performa dalam keadaan Regular
3G**

| No | Waktu Pengiriman <i>Server</i> (Jam:menit:detik) | Waktu Penerimaan <i>Server</i> (Jam:menit:detik) | Waktu Selisih(detik) |
|-----------|--|--|----------------------|
| 1 | 20:39:42 | 20:40:42 | 60 |
| 2 | 20:42:04 | 20:42:55 | 51 |
| 3 | 20:43:12 | 20:43:58 | 46 |
| 4 | 20:45:44 | 20:46:33 | 49 |
| 5 | 20:48:14 | 20:49:01 | 47 |
| 6 | 20:49:25 | 20:50:14 | 48 |
| 7 | 20:50:28 | 20:52:17 | 109 |
| 8 | 20:52:34 | 20:53:31 | 57 |
| 9 | 20:53:45 | 20:55:32 | 107 |
| 10 | 21:05:49 | 21:07:43 | 114 |
| Rata-rata | | | =68.8 |

Tabel 5.10 Hasil Uji Coba Performa dalam keadaan GPRS

| No | Waktu Pengiriman <i>Server</i> (Jam:menit:detik) | Waktu Penerimaan <i>Server</i> (Jam:menit:detik) | Waktu Selisih(detik) |
|-----------|--|--|----------------------|
| 1 | 21:16:45 | 21:31:31 | 886 |
| 2 | 21:33:48 | 21:42:47 | 549 |
| 3 | 21:43:05 | 21:51:05 | 480 |
| 4 | 21:51:42 | 21:59:07 | 445 |
| 5 | 21:59:34 | 22:07:42 | 488 |
| 6 | 22:10:09 | 22:19:12 | 543 |
| 7 | 22:20:37 | 22:28:48 | 491 |
| 8 | 22:43:50 | 22:56:31 | 761 |
| 9 | 23:02:37 | 23:10:37 | 480 |
| 10 | 23:15:02 | 23:29:53 | 891 |
| Rata-rata | | | =596.8 |

Setelah itu dilakukan juga uji coba performa yang membandingkan proses metode pengunggahan yang terdapat pada tugas akhir ini yaitu dengan cara memotong-motong *file* yang dikirim terlebih dahulu dengan metode pengiriman *file* normal yang dilakukan tanpa melakukan proses pemotongan *file* terlebih dahulu. Setelah diadakan uji coba tersebut hasilnya metode

pengiriman file menggunakan metode pemotongan file rata rata memiliki proses lebih lama yaitu 32 detik sedangkan apabila tanpa proses pemotongan file dapat dikirim ke *server* dalam waktu rata-rata 21 detik. Proses pengiriman file menggunakan metode pemotongan file memakan waktu 51% lebih lama dibanding metode pengiriman file biasa. Hasil uji Coba dapat dilihat pada tabel berikut ini:

Tabel 5.11 Hasil Uji Coba Upload Menggunakan Metode *Chunk File*

| No | Waktu Pengiriman <i>Server</i> (Jam:menit:detik) | Waktu Penerimaan <i>Server</i> (Jam:menit:detik) | Waktu Selisih(detik) |
|----|---|---|----------------------|
| 1 | 12:09:46 | 12:10:14 | 28 |
| 2 | 12:11:00 | 12:11:58 | 58 |
| 3 | 12:15:26 | 12:15:58 | 32 |
| 4 | 12:17:17 | 12:17:47 | 30 |
| 5 | 12:18:37 | 12:19:02 | 25 |
| 6 | 12:22:12 | 12:22:40 | 28 |
| 7 | 12:23:38 | 12:24:38 | 60 |
| 8 | 12:25:30 | 12:26:05 | 35 |
| 9 | 12:28:57 | 12:29:28 | 31 |
| 10 | 12:33:10 | 12:33:41 | 31 |

| | | | |
|-----------|--|--|-------|
| Rata-rata | | | =32.8 |
|-----------|--|--|-------|

Tabel 5.12 Hasil Uji Coba Upload Menggunakan Metode Upload PHP

| No | Waktu Pengiriman <i>Server</i> (Jam:menit:detik) | Waktu Penerimaan <i>Server</i> (Jam:menit:detik) | Waktu Selisih(detik) |
|-----------|--|--|----------------------|
| 1 | 12:09:45 | 12:10:11 | 26 |
| 2 | 12:11:02 | 12:11:21 | 19 |
| 3 | 12:15:27 | 12:15:47 | 20 |
| 4 | 12:17:16 | 12:17:42 | 26 |
| 5 | 12:18:39 | 12:19:59 | 20 |
| 6 | 12:22:13 | 12:22:34 | 21 |
| 7 | 12:23:39 | 12:24:00 | 21 |
| 8 | 12:25:28 | 12:26:53 | 25 |
| 9 | 12:28:56 | 12:29:16 | 20 |
| 10 | 12:33:11 | 12:33:34 | 23 |
| Rata-rata | | | =21.5 |

(Halaman ini sengaja dikosongkan)

BAB VI PENUTUP

Pada bab ini akan dibahas mengenai mengenai kesimpulan yang dapat diambil dari tujuan rancang bangun aplikasi serta hasil coba yang telah dilakukan pada Tugas Akhir ini. Selain itu juga terdapat beberapa saran untuk pengembangan aplikasi lebih lanjut.

6.1. Kesimpulan

Berdasarkan hasil pengamatan, perancangan, implementasi dan uji coba aplikasi, maka dapat diambil beberapa kesimpulan dari hasil pembuatan tugas akhir ini, yaitu:

1. Pemotongan file tugas bersifat adaptif dengan kecepatan pengguna, server dapat menerima file sesuai dengan yang dikirim pengguna
2. Pada uji performa pada keadaan normal tanpa latensi server dapat melakukan proses dengan cepat yaitu sekitar 29 detik, pada kondisi jaringan 3g proses terjadi selama 68 detik sedangkan pada kondisi GPRS proses terjadi selama 596 detik.
3. Pada uji performa pengiriman file metode pemotongan, didapatkan bahwa kecepatan pengiriman file metode pemotongan file lebih lambat 51% dibandingkan dengan metode tanpa pemotongan
4. Waktu yang dibutuhkan metode pemotongan *file* lebih lama karena terdapat proses penggabungan kembali file yang sudah terkirim pada *server*
5. Aplikasi dapat melanjutkan pengiriman *file* apabila jaringan terputus saat proses pengiriman *file*.

6.2.Saran

Berdasarkan hasil pengamatan, perancangan, implementasi, serta hasil uji coba aplikasi ini, maka diperlukan beberapa saran untuk pengembangan aplikasi lebih lanjut, yaitu:

1. Perlunya klasifikasi kecepatan unggah yang lebih baik agar aplikasi lebih adaptif terhadap keragaman jaringan yang dimiliki pengguna.
2. Diperlukan metode lebih baik untuk mengirim *file*, sebab metode ini memakan waktu lebih lama dibanding metode pengiriman file biasa.

LAMPIRAN

1. Kode index.php untuk fungsi upload

```
<?php

include_once "../lib/php/db-config.php";

include_once "config.php";

$id_user = $_SESSION['id_user'];

$id_kelas = $_SESSION['id_kelas'];

$id_tugas = $_SESSION['id_tugas'];

$kembali = $_SESSION['kembali'];

$q = mysqli_query($conn, "    SELECT id FROM
kelas_tugas_pengumpulan WHERE id_kelas
='$id_kelas' AND id_tugas    = '$id_tugas' AND
id_peserta_didik = '$id_user'");

$n = mysqli_num_rows($q);

if($n == 1){

    $_SESSION['keperluan']="Re-Upload tugas";

}else{

    $_SESSION['keperluan'] = "Upload tugas";

}

$keperluan = $_SESSION['keperluan'];
```

```
$folder="../../student/tugas/files/USR_".$id_user.  
"/KLS_".$id_kelas."/TGS_".$id_tugas;  
  
//echo $folder;  
  
if(is_dir($folder)) {  
  
echo ("Folder sudah ada");  
  
}  
  
else {  
  
    if(mkdir($folder, 0777, true)){  
  
        // echo 'Folder baru telah di buat';  
  
    }  
  
    else {  
  
        // echo ("Tidak bisa membuat folder baru");  
  
    }  
  
}  
  
$qCekTugas = "SELECT `nm_tugas`, `deskripsi`  
FROM `kelas_tugas` WHERE `id`='". $id_tugas. "'";  
  
$exeCekTugas = mysqli_query($conn,$qCekTugas);  
  
$dtTugas = mysqli_fetch_array($exeCekTugas);  
  
$nm_tugas = $dtTugas['nm_tugas'];  
  
$deskripsi = $dtTugas['deskripsi'];  
  
?>
```



```
<!DOCTYPE html>

<head>

<script type="text/javascript">

var timerStart = Date.now();

</script>

<!-- Standard Meta -->

<meta charset="utf-8" />

<meta http-equiv="X-UA-Compatible"
content="IE=edge,chrome=1" />

<meta name="viewport" content="width=device-width,
initial-scale=1.0, maximum-scale=1.0">

<!-- Site Properities -->

<title>Upload tugas</title>

<link rel="stylesheet" type="text/css"
href="./style2/semantic.css">

<script src="./style2/jquery.min.js"></script>

<script src="./style2/semantic.js"></script>

</head>

<body>

<div class="ui blue inverted segment">

<h3 align="center"><?php echo $keperluan; ?> <br/>
<?php echo $nm_tugas; ?></h3>

<p align="center"><?php echo $deskripsi; ?></p>
```

```

</div>

<div class="ui segment">

<div class="ui grid">

<div class="twelve wide stretched column">

<div class="ui segment" id="viewResult">

</div>

</div>

<div class="four wide column" id="menuList">

<div class="ui vertical fluid right tabular menu">

<a class="active item" id="fileUpload"
onClick="loadFile('file.php','fileUpload')">

<i class="file icon"></i> Upload Dari File </a>

<a class="item" id="photo"
onClick="loadFile('camera.php','photo')">

<i class="photo icon"></i> Simpan Dari Kamera </a>

<a class="item" id="uncompleted"
onClick="loadFile('uncompleted.php','uncompleted')
">

<i class="repeat icon"></i> Uncompleted Upload
</a>

<a class="item" id="myFile"
onClick="loadFile('myfile.php','myFile')">

<i class="file text icon"></i> File Saya </a>

</div></div></div></div>

```

```

<div class="ui ignored info message"><i
class="history icon"></i><i>Time Execution:

```

```

</i><span id="speedTest">5</span> <b><i>(Upload:
<span
id="speedResult">Unknow</span></i></b></div>

<div class="ui segment">

<button class="circular ui icon orange button"
onClick="kembali();">

<i class="icon arrow circle left"></i> Kembali Ke
dashboard </button>

</div><div id="speedTest"
style="visibility:hidden"></div></body>

</html>

<script type="text/javascript">

    function loadFile(url,active){

        $("a").removeClass('active');

        $("#"+active).addClass('active');

        $("#viewResult").load(url);

    }$("#viewResult").load("file.php");

    function kembali(){

        window.location.href = "<?php echo
$_SESSION['kembali']; ?>";

    }window.onload = function(){

        setTimeout(function(){

            var t = performance.timing;

            console.log(t.loadEventEnd -.responseEnd);

            $("#speedTest").text(t.loadEventEnd -
t.responseEnd);

```

```
}, 0);
```

```
}</script>
```

DAFTAR PUSTAKA

- [1] “CMS evaluation guide,”[Online].Available:
https://omniupdate.com/_resources/pdfs/white_papers/cms_evaluation_guide.pdf
- [2] “Chunked *File* Upload,”[Online]Available:
<https://github.com/blueimp/jQuery-File-Upload/wiki/Chunked-file-uploads>
- [3] H. Xinhua dan H. Wenfa, “An Innovative Web-Based Collaborative Learning Model and Application Structure,” 2008.
- [4] R. Zhao dan C. Zhang, “A Framework for Collaborative Learning System Based on Knowledge,” 2009.
- [5] “MySQL :: The world's most popular open source database,” [Online]. Available:
<http://www.mysql.com/>.
- [6] “PHP: Hypertext Preprocessor,” [Online]. Available:
<http://php.net/>.
- [7] “Javascript: Ready to try Javascript,” [Online]. Available: <https://www.javascript.com/>.
- [8] Batra, S. (2003). *AJAX - Asynchronous Java Script and XML*. Salzburg: Information Technology and System Management

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Rahmat Daniyanto, lahir di kota Jakarta, pada tanggal 21 Agustus 1993. Penulis adalah anak pertama dari dua besaudara. Pendidikan formal yang telah ditempuh penulis adalah TK Puspita Mekar (1997-1999), SD 05 Malaka Sari Jakarta Timur(1999-2005), SMPN 255 Jakarta (2005-2008), SMA 12 Jakarta (2008-2011). Setelah lulus dari SMA

12 Jakarta, penulis diterima di jurusan Teknik Informatika ITS angkatan 2011 dengan NRP 5111100149. Penulis dapat dihubungi melalui alamat *e-mail* di rdani012@gmail.com.